Chapter 8: Fixity

Topics covered in this chapter:

GLOSSARY:

copy: An instance of a package stored in a silo pool. For any given package, there will be at most one copy per silo pool.

silo pool database (store_db): The relational database in which the Silo Pool Service keeps track of the packages stored within itself. There is a silo pool database for every silo pool, and the silo pool database contains expected and last calculated fixity data for packages stored in its constituent silos. Additionally, historical information is stored for each package: the times of initial storage, all fixity checks, time of deletion, and when a package is missing.

storage master database (daitss_db.storemaster): The relational database in which the Storage Master Service keeps track of copies of packages across all silo-pool instances. There is one storage master database in the DAITSS 2 storage system.

DAITSS preservation database (daitss_db.daitss2)**:** The relational database in which the DAITSS Core Service stores preservation and operational metadata, including fixity data, for all packages ingested in the archive.

disk-fixity: A utility, part of the Silo Pool Service, that calculates the fixity of all packages on a disk silo pool and updates the silo pool database.

collect-fixities: A utility, part of the Storage Master service, that collects all the fixity data for all silo pools and compares it to the fixity data in the DAITSS preservation database. It writes the result of this comparison to the DAITSS preservation database.

fixity expired time: The age past which a fixity check is considered too old to be valid.

fixity stale time: The age past which a fixity check is ready to be recalculated.

fixity success: The state of a package for which all expected copies of the package are present and whose checksums match each other and the DAITSS preservation database.

fixity failure: The state of a package for which one or all copies have checksums that are not consistent with each other or the DAITSS preservation database.

integrity failure: The state of a package for which one or more copies are missing.

fresh_enough time: A fixity check performed within this time will not be recalculated.

Overview

DAITSS fixity checking can be broken down into two sub-processes: silo-side fixity computation and reconciliation.

Silo-side fixity computation

The process of silo-side fixity computation is implemented by the disk-fixity utility and runs independently on each silo pool. It has three functions:

1. Compute new fixity data (md5 and sha1 checksums) for each copy stored in the silo pool. If a package has been checked more recently than the fresh_enough time, the fixity is not recomputed.

2. Update the silo pool database so that it always reflects the most recent fixity data.

3. Report any inconsistencies found. These inconsistencies include the following conditions:

- **fixity failure**: The computed checksum for a package doesn't match the expected checksum.
- **missing package**: A package expected to be in the silo pool is not present.
- **alien package**: A package is found on the silo that isn't accounted for in the silo pool database.
- **ghost package**: A package that the silo database says is deleted is present in the silo.

The process of silo-side fixity computation is implemented by the disk-fixity script, which should be scheduled for periodic execution using cron.

Reconciliation

The process of reconciliation is implemented by the collect-fixities utility. It has five functions:

1. For each package in the archive, determine whether the checksums stored for each copy match each other and the checksums stored for the package in the DAITSS preservation database.
2. For each package in the archive, determine whether the required number of copies exist.
3. For each package in the archive, determine whether any of the packages are completely missing, having no copies at all.
4. For each package in the archive, update the package's DAITSS preservation database fixity events to reflect the outcome of the determinations described above.
5. Issue a report summarizing the results of the reconciliation.

The following table illustrates these comparisons and the package state that would be written to the DAITSS preservation database fixity events for an example set of packages with two required copies:

	silo pool 1 checksum	silo pool 2 checksum	daitss preservation database checksum	package state
package a	x	x	x	fixity success
package b	x	y	x	fixity failure
package c	y	y	x	fixity failure
package d	missing	x	x	integrity failure
package e	missing	missing	x	Integrity failure

Running the DAITSS fixity utilities

disk-fixity

On the DAITSS demonstration system, disk-fixity can be found in the directory /opt/web-services/sites/silo-pool/tools/disk-fixity.

Disk-fixity expects the following configuration directives to be defined in the daitss-config.yml file under the disk-fixity section:

Directive	Purpose
log_syslog_facility	Optionally, a syslog facility to send log messages to.
log_filename	Optionally, the path to a file to write log messages in. At least one of log_syslog_facility or log_filename should be defined.
Hostname	HTTP hostname of the silo-pool to be checked. This does not include a port number.
pid_directory	Directory path in which to write the process id file.
fresh_enough	Age in days of the most recent fixity check for a package before which a fixity check will be skipped.

Configuration:

Disk-fixity expects the following configuration directives to be defined in the daitss-config.yml file under the defaults section:

Directive	Purpose
fixity_expired_days	Time in days after which a fixity check is considered to be expired.
fixity_stale_days	Time in days after which a fixity check is considered to be stale and should be recalculated.

Running disk-fixity:

To execute disk-fixity, type the following commands into the terminal:

```
cd /opt/web-services/sites/silo-pool

bundle exec tools/disk-fixity &
```

After disk-fixity finishes a pass on all copies in the silo-pool, it will write a summary report to standard output and terminate.

The output and report generated by disk-fixity can be seen in the disk-fixity log, located in /var/log/daitss/daemons/disk-fixity.log.

On the DAITSS demonstration system, disk-fixity is scheduled to run daily at 1:01 AM via cron. It is not necessary to run disk-fixity manually. The summary report will be mailed to the root user.

collect-fixities

On the DAITSS demonstration system, collect-fixities can be found in the directory /opt/web-services/sites/storage-master/tools/collect-fixities.

Configuration:

Collect-fixities expects the following configuration directives to be defined in the daitss-config.yml file under the collect-fixities section:

Directive	Purpose
log_syslog_facility	Optionally, a syslog facility to send log messages to.
log_filename	Optionally, the path to a file to write log messages in. At least one of log_syslog_facility or log_filename should be defined.
pid_directory	Path of directory to write process id file to.

Directive	Purpose
server_address	HTTP address of the Storage Master service, including the port number if it is not the standard port 80.

Collect-fixities expects the following configuration directives to be defined in the daitss-config.yml file under the defaults section:

Directive	Purpose
required_pools	Number of copies required per package. There must at least be this many solo pools available.

Running collect-fixities:

To execute collect-fixities, type the following commands into the terminal:

```
cd /opt/web-services/sites/store-master

bundle exec tools/collect-fixities &
```

After collect-fixities finishes a pass on all copies in the archive, it will write a summary report to standard output and terminate.

The output and report generated by collect-fixities can be seen in the collect-fixities log, located in /var/log/daitss/daemons/collect-fixities.log.

On the DAITSS demonstration system, collect-fixities is scheduled to run daily at 2:02 AM via cron. It is not necessary to run collect-fixities manually. Its summary report will be mailed to the root user.