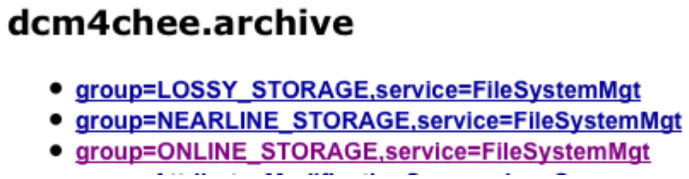
Overview of the Amazon S3 HSM Module

This document will provide a brief overview of the dcm4chee nearline implementation as relates to the Amazon S3 plugin.

NEARLINE vs. ONLINE

In dcm4chee you can set up multiple file systems. This is represented by FileSystemMgt MBeans which can have multiple file systems (mount points or drives) associated with them. They are designated as groups. This is represented in the JMX Console in the following way:



dcm4chee.archive

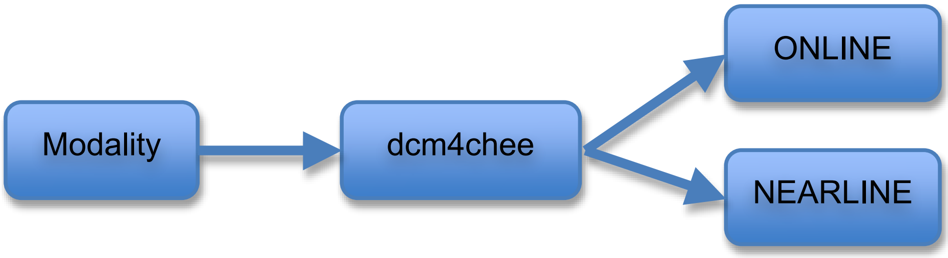
- group=LOSSY_STORAGE,service=FileSystemMgt
- group=NEARLINE_STORAGE,service=FileSystemMgt
- group=ONLINE_STORAGE,service=FileSystemMgt

By default, an ONLINE, NEARLINE and a LOSSY storage group exist. We are not going to worry about the LOSSY group in this document.

The thought behind these two file system groups is that online storage represents fast, direct access storage. Nearline storage is expected to be (most likely) cheaper storage with slower performance. It is used as a secondary or disaster recovery copy, or simply as a way to tier storage and reduce costs by storing less frequently accessed studies on a cheaper storage system.

In the default configuration, the nearline file system in dcm4chee is not configured to do anything. It is inactive.

When a NEARLINE file system is activated, the storage will look like this:



Each file system group may be managed separately. One of the benefits of this is that you may specify different deletion criteria for each group. So, you could set up a rule that deletes the files from online if they have not been accessed for a year and they exist in an ARCHIVED state on the NEARLINE file system group.

In dcm4chee, there is a pluggable abstraction layer for handling the file system interaction with nearline storage. A lot of types of nearline storage require direct hardware or software interfaces. Because it is not practical for most application vendors to develop drivers for every type of nearline storage device (tape, CD, DVD, etc.) a market has been create for software layer storage managers. These are called Hierarchical Storage Managers, or HSM. An HSM is usually responsible for watching a file system and synchronizing its state with a nearline device. Because there are different types of HSM as well, the plugin model of dcm4chee nearline storage is focused around HSM concepts. The default HSM modules can either directly interact with a file system (relying on the HSM to do everything) or perform all interactions by invoking a command line interface. We will take advantage of this HSM plugin model to implement the S3 storage interface.

Archiving to Amazon S3 as a Nearline File System

Amazon S3 is a cloud-based storage system. A client makes use of an API (with an underlying HTTP protocol) to communicate with S3. Objects are stored within S3, and referenced via a string key.

Add the files

While the system is shut down, add the following files to your dcm4chee installation in the specified places:

- dcm4chee-hsm-s3.jar file to /server/default/lib
- dcm4chee-hsm-s3-module-xmbean.xml to /server/default/conf/xmdesc
- dcm4chee-hsm-s3-module-service.xml to /server/default/deploy

The jar file contains the code for the service, the xmbean XML file contains the service descriptor, and the service XML file contains the deployment descriptor.

After these files have been added, start dcm4chee.

Configure the NEARLINE storage group and S3 HSM module

In this section we will configure the various services and get an understanding of what they do.

1) Add a file system to the NEARLINE storage group

- Open a web browser, and navigate to the JMX console. e.g. http://localhost:8080/jmx-console
- Locate the “dcm4chee.archive” section, and click on “group=NEARLINE_STORAGE,service=FileSystemMgt”
- Scroll down to the “List of MBean Operations” section and find the “addRWFileSystem()” operation. Enter in a path for nearline storage. This will not actually be used, since we are going to store the files in S3, but we need to configure something here so that the system knows we are using the NEARLINE storage. I have entered: “tar:/storage/nearline”. Note the *tar* prefix. This tells dcm4chee that all of the files going to this storage group will be tarred up.
- Click Invoke to create add the file system record into the database.

2) Configure the FileCopy service

The FileCopy service is responsible for physically copying files to your nearline storage. This is where we configure our particular plugin.

- Specify a value for the DestinationFileSystem. This value should equal the value you specified for your nearline storage file system so that dcm4chee knows that this FileCopy service is associated with that file system configuration. e.g. “tar:/storage/nearline”
- Specify a value for HSMModulServicename. This should be the JMX ObjectName of our S3 plugin module, and enables it for use within this service when storing and retrieving files. Enter: “dcm4chee.archive:service=FileCopyHSMModule,type=S3”
- Leave the FileStatus set to TO_ARCHIVE. This will be the status of files stored in S3. When the SyncFileStatus service runs and verifies that these files are stored properly, it will change the status to ARCHIVED.
- Click Apply Changes.

3) Configure the TarRetriever service

This service is responsible for fetching and extracting tar files from the nearline storage during retrieve requests.

- Specify a value for HSMModulServicename. This should be the JMX ObjectName of our S3 plugin module, and enables it for use within this service when storing and retrieving files. Enter: “dcm4chee.archive:service=FileCopyHSMModule,type=S3”
- Click Apply Changes.

4) Configure the S3 HSMModule (service=FileCopyHSMModule,type=S3)

Now we are ready to configure the S3 integration. Here are the main things to configure here:

- Amazon S3 bucket name
- Amazon AWS Access Key
- Amazon AWS Secret Key (this is write only, and you will not see a value after clicking Apply Changes)
- The Outgoing and Incoming directories in this configuration are temporary storage areas that are used for tarring and untarring files.
- Click Apply Changes.

5) Configure the SyncFileStatus service

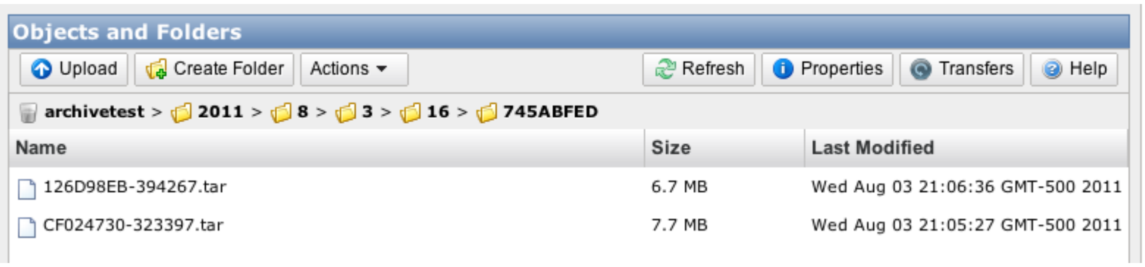
This service will run periodically and verify the files that have been stored to S3. It will fetch the tar files and ensure that the correct files are contained within. Once it verifies the files, it will update the file status in the database to ARCHIVED.

- Specify a value for the MonitoredFileSystem. This value should equal the value you specified for your nearline storage file system so that dcm4chee knows that this service is associated with that file system configuration. e.g. “tar:/storage/nearline”

- Specify a value for HSMModulServicename. This should be the JMX ObjectName of our S3 plugin module, and enables it for use within this service when fetching tar files. Enter: “dcm4chee.archive:service=FileCopyHSMModule,type=S3”
- Specify a TaskInterval. It is set to NEVER by default, so you you should set it to a proper interval that is good for your workflow, preferably not during peak business hours.

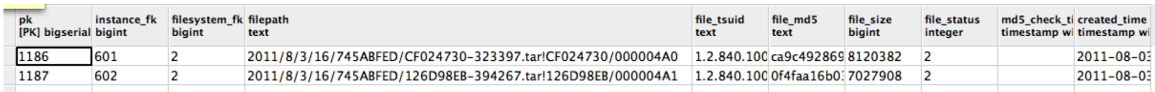
Summary

At this point you should be able to store DICOM objects to dcm4chee and it will archive them to Amazon S3. The S3 key will be a hierarchical path, which should look familiar to you if you have looked at how dcm4chee stores objects on a file system. For example, here is a screenshot of my Amazon Management Console showing the archived path:



Name	Size	Last Modified
126D98EB-394267.tar	6.7 MB	Wed Aug 03 21:06:36 GMT-500 2011
CF024730-323397.tar	7.7 MB	Wed Aug 03 21:05:27 GMT-500 2011

In the database, the files should have a changed file status, and should reflect their tar path as shown in this screenshot:



pk [PK] bigserial	instance_fk bigint	filesystem_fk bigint	filepath text	file_tsuid text	file_md5 text	file_size bigint	file_status integer	md5_check_ti timestamp wi	created_time timestamp wi
1186	601	2	2011/8/3/16/745ABFED/CF024730-323397.tar!CF024730/000004A0	1.2.840.100	ca9c492869	8120382	2		2011-08-03
1187	602	2	2011/8/3/16/745ABFED/126D98EB-394267.tar!126D98EB/000004A1	1.2.840.100	0f4faa16b0	7027908	2		2011-08-03

If you can’t read it in the picture, the filepaths look like this:

```
2011/8/3/16/745ABFED/CF024730-323397.tar!CF024730/000004A0
```

Note the “tar” designator in the path. This tells the system that the file is contained within a tar file.

Setting up Retention Rules to Remove Studies from Online

We probably don’t want two copies of the study forever, so lets set up some rules now so that studies are deleted from the ONLINE storage group after a period of time. This will leave the remaining copy on S3.

Note that this is only an example. Your retention/deletion requirements may differ!

1) Configure deletion of ONLINE studies

- Open a web browser, and navigate to the JMX console. e.g. http://localhost:8080/jmx-console
- Locate the “dcm4chee.archive” section, and click on “group=ONLINE_STORAGE,service=FileSystemMgt”

- Set DeleteStudyIfNotAccessedFor = your retention period (52w or whatever your SLA requires)
- Set DeleteStudyOnlyIfStorageNotCommited = false
- Set DeleteStudyOnlyIfCopyOnMedia = false
- Set DeleteStudyOnlyIfCopyOnReadOnlyFileSystem = false
- Set ScheduleStudiesForDeletionInterval = a reasonable time interval for the system to check the database and schedule deletion jobs.
- Set DeleteStudyOnlyIfCopyOnFileSystemOfFileSystemGroup = NEARLINE_STORAGE
- Set DeleteStudyOnlyIfCopyArchived = true (only delete studies that have been verified by the SyncFileStatus service. If you don't care about that or are not running that service, you can set this false.)
- Click Apply Changes

At this point, dcm4chee will look for studies in ONLINE that meet these criteria and schedule them for deletion. After they are deleted, and the only copy is on S3, a retrieve request will trigger a fetch from Amazon. The tar file(s) will be fetched, images extracted and sent to the destination.

That's it!