

**DataFrameworks**

TECHNOLOGY SUMMARY

## **ClarityNow! Plug-Ins**

---

## Table of Contents

<b>Amazon S3 Plugin</b> .....	<b>3</b>
<b>Archive Plugin</b> .....	<b>3</b>
<b>ASG / Atempo ADA Plugin</b> .....	<b>4</b>
<b>Audited Delete Plugin</b> .....	<b>4</b>
<b>DMF Copy Plugin</b> .....	<b>4</b>
<b>SIG DMF "dmls" Plugin</b> .....	<b>5</b>
<b>Duplicate Plugin</b> .....	<b>5</b>
<b>Filtered Archive Plugin</b> .....	<b>6</b>
<b>Media Info Plugin</b> .....	<b>7</b>
<b>Object Interface Plugin</b> .....	<b>7</b>
<b>Pixspan Compression Plugin</b> .....	<b>8</b>
<b>Preview Plugin</b> .....	<b>8</b>
<b>Recursive File Listing Plugin</b> .....	<b>8</b>
<b>Restore Plugin</b> .....	<b>9</b>
<b>ScheduAll Plugin</b> .....	<b>9</b>

## Amazon S3 Plugin

Intended to provide data creators / consumers (Producers, Primary Investigators, Research Scientist, etc) “self-service” ability to send data to Amazon S3 Bucket.

- POC status with movement from file system into Amazon S3 via native S3 interface
- Returns both the AWS eTag and the AWS Link (URL)
- Includes audit trail
- Creates mirror of the file system naming conventions within S3 bucket



Fig. 1.1 — AWS Archiving interface



Fig. 1.2 — Amazon S3 plugin interface

## Archive Plugin

Intended to provide data creators / consumers (Producers, Primary Investigators, Research Scientist, etc) “self-service” ability to send data to an archive system.

- Copy data from source file system to destination archive file system
- Includes audit trail
- Uses python utility (equivalent to unix cp) and commands to accomplish the copy of data to the archive
- Some flexibility on destination naming conventions

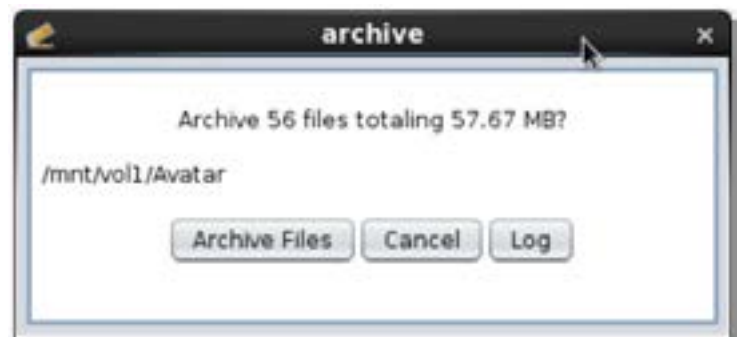


Fig. 2 — Archive plugin dialogue window

## ASG / Atempo ADA Plugin

Designed to automate the transfer of meta data from ClarityNow! into the ADA archive software reducing / removing the need to manual meta data entry into ADA

- POC status supports seamless transfer of meta data from ClarityNow! to ADA
- Creates an XML file used by ADA to ingest meta data
- ClarityNow! provides plugin search button to pass search criteria to plugin. Such functionality could be used to query the ADA repository from within ClarityNow! interface

## Audited Delete Plugin

Intended to provide data creators / consumers (Producers, Primary Investigators, Research Scientist, etc) “self-service” ability to delete data from predetermined areas on file systems without actually granting users file system access.

- Provides audit trail logging

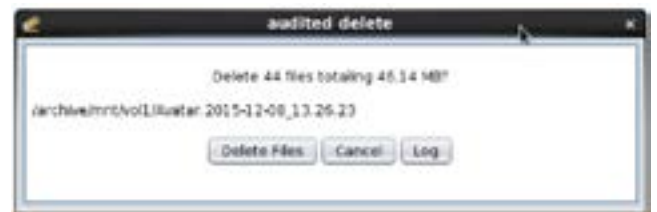


Fig. 3 — Audited Delete plugin dialog window

## DMF Copy Plugin

Intended to provide data creators / consumers (Producers, Primary Investigators, Research Scientist, etc) “self-service” ability to send data to an archive system. Designed for large data sets and therefore uses high-speed, parallel copy mechanisms.

- Moves or copies data from source file system to destination filesystem
- Supports any source / destination file system that is mounted on data mover
- Includes audit trail
- Supports multiple parallel data movers (cxfsdp, MCP)
- Supports remote data movers (specify remote machines dedicated to data movement and checksum operations)
- Supports MD5 checksum validation
- Supports trial run (to pre identify issues such as invalid characters, etc.)
- No flexibility to customize on naming conventions on destination archive



Fig. 4 — DMF Copy plugin interface



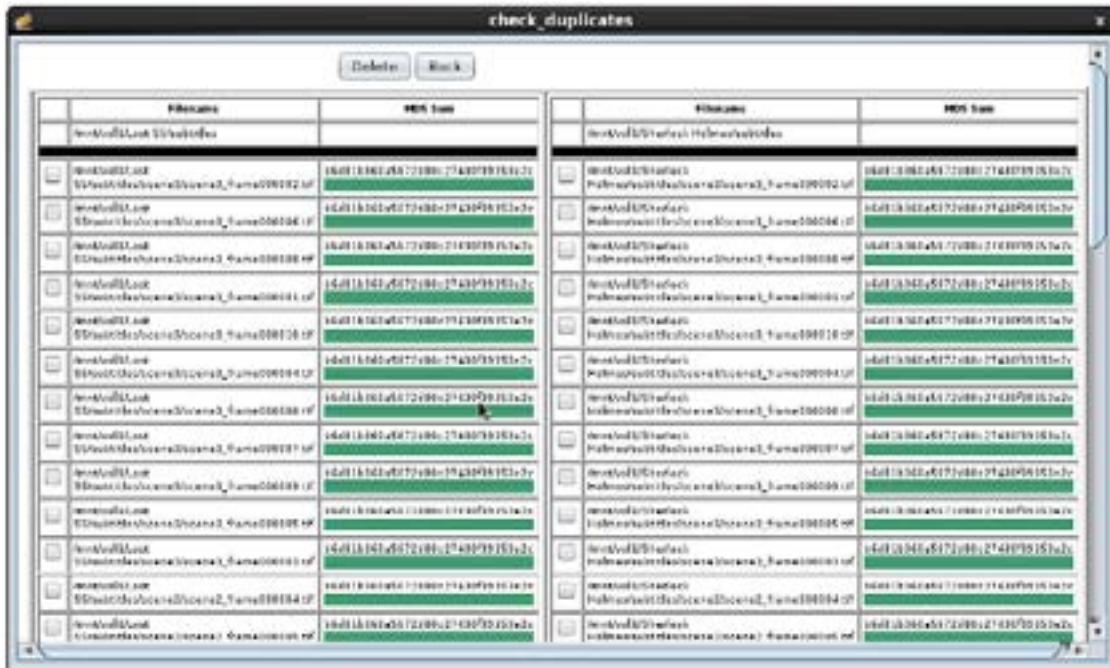


Fig. 5.2 — Duplicate plugin interface

## Filtered Archive Plugin

The Filtered Archive Plugin is automated stand-alone python module designed to provide filter list of directories to be archived as input to a data mover module. The Filtered Archive Plugin provides the following functionality:

- Executes on daily (or configurable) basis via cron
- Obtains list of folders from ClarityNow! with “expired” status
- Filters list for “expired” folders that have surpassed the allowable period for “pending archive” postponement requests (it is assumed to be 7 days but period is configurable)
- Filter for those meeting archiving requirements (location, naming)
- Creates an output file with filtered list in %pfile format that is used as input to actual data mover. The %pfile format enables ability to use “stand-alone” data mover plugin via command line
- Invokes data mover to execute data movement

## Media Info Plugin

The MediaInfo plugin is designed to provide media attribute meta data attributes about a file without granting the ClarityNow! user any read / write file system access.

- MediaInfo (<https://mediarea.net/en/MediaInfo>) open source packages to generate meta data window



Fig. 6 — Media Info plugin interface

## Object Interface Plugin

Intended to provide data creators / consumers (Producers, Primary Investigators, Research Scientist, etc) “self-service” ability to send data to the Scality object store.

- POC status with movement from file system into object storage via HTTP (Get, Put, Delete) interface
- Currently works with the Scality sproxyd HTTP interface
- Returns both the Object ID and the Object URL
- Includes audit trail
- Creates mirror of the file system naming conventions within Scality URL path using the “by path” bpchord functionality within the Scality object store



Fig. 7.1 — Send to Scality dialogue window



Fig. 7.2 — Scality archive interface

## Pixspan Compression Plugin

Intended to provide data creators / consumers (Producers, Primary Investigators, Research Scientist, etc) “self-service” ability to send data to an archive system. Designed to invoke the Pixspan compression technology during the archive process.

- Pixspan compression technology compresses high resolution uncompressed image files (cin, dpx, exr, etc.)
- Supports any source / destination file system that is mounted on data mover
- Includes audit trail
- Supports MD5 checksum validation
- Some flexibility to customize on naming conventions on destination archive



Fig. 8 — Pixspan Compression interface

## Preview Plugin

Preview plugin is designed to provide a thumbnail view and media info meta data attributes about a file without granting the ClarityNow! user any read / write file system access.

- Uses ImageMagick (<http://www.imagemagick.org>), ffmpeg ([www.ffmpeg.org](http://www.ffmpeg.org)), and MediaInfo (<https://mediainfo.net/en/MediaInfo>) open source packages to generate preview window.



Fig. 9 — Preview plugin interface

## Recursive File Listing Plugin

The Recursive File Listing Plugin is designed to find files within the selected directory and all subdirectories that have not been changed over some X period of time. The resulting list is copied to the client clipboard so the results can be pasted into a spreadsheet, etc. for further analysis.



- Copies list of files with creation date greater than X days old to clipboard
- Uses Unix “ctime” do identify files to determine the date a file was last changed. The ctime gets updated when the file attributes are changed, like changing the owner, changing the permission or moving the file to an other filesystem but will also be updated when you modify a file.
- Formats the data “tab separated” for pasting into spreadsheet

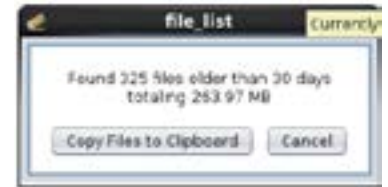


Fig. 10 — File list dialogue window

## Restore Plugin

Intended to provide data creators / consumers (Producers, Primary Investigators, Research Scientist, etc) “self-service” ability to restore data from an archive system.

- Moves data from archive file system back to source file system
- Includes audit trail
- Uses python utility (equivalent to unix cp) and commands to accomplish the copy of data from the archive
- Some flexibility to customize resulting naming conventions on restored file system

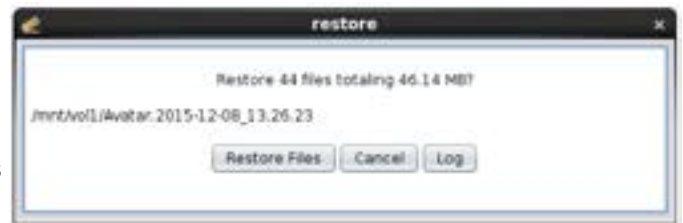


Fig. 11 — Restore plugin dialogue window

## ScheduAll Plugin

The ScheduAll plugin uses the ClarityNow! tag functionality to harvest additional meta data from the ScheduAll “work order” table (example: Producer, Client, Content Owner, etc). and apply as ClarityNow! tags. Also capable of pushing storage cost information into field within ScheduAll

- Supports harvest of additional meta data from ScheduAll work order table to apply as ClarityNow! implied tags
- Needs to be customized for each ScheduAll customer



# DataFrameworks

© 2016 DataFrameworks, Inc.  
5737 Kanan Road, Suite 308, Agoura Hills, CA 91301  
(925) 201-5123  
dataframeworks.com

DataFrameworks is a registered trademark of DataFrameworks, Inc. XYZ, Blah, and Something are trademarks of Dataframeworks, Inc. All other company and product names used herein may be the trademarks of their respective companies. Company and the Company logo are trademarks of Company, Inc., in the United States and/or other countries.