MCPClient

From Archivematica

Main Page > Development > Development documentation > MCPClient

Archivematica has one or more MCPClient instances to perform the actual work. They are gearman worker implementations that inform the gearman server what tasks they can perform, and wait for the server to assign them a task. When a client starts, it connects to the specified gearman server and provides a list of modules they support. When the MCPServer informs the gearman server of a Task that the client supports and the gearman server assigns the job to the client, the client will process the Job, and return the results to the gearman server, which in turn will return them to the MCPServer.

Design
This page proposes a new feature and reviews design options

Development
This page describes a feature that's in development

Documentation
This page documents an implemented feature

Contents

Client scripts

Client scripts do the actual work in Archivematica. They are anything that can be run on the command line, from builtins like mv and cp, to custom-written scripts.

New scripts are defined in `src/MCPClient/lib/archivematicaClientModules (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/archivematicaClientModules)` , which is what is registered with Gearman on MCPClient startup.

Improvement note: archivematicaClientModules lists both 'supportedCommandSpecial' and 'supportedCommands'. This distinction may have once been based on scripts that relied on external services, but serves no purpose now and should be removed.

The name is what the StandardTasksConfig table will refer to them as, and the value is the script that will be run. Some are defined as shell builtins (eg copy_v0.0 is cp). Most are paths to a script in the clientScripts directory, using the `%clientScriptsDirectory%` replacement variable. The name of the client script is usually the same as the name in archivematicaClientModules, but for very old scripts may have ‘archivematica’ at the beginning (eg createMETS_v2.0 = archivematicaCreateMETS2.py) or be named more pythonically (eg parseExternalMETS = parse_external_mets.py). Entries are added alphabetically.

The version (eg copy_v0.0) was originally intended to be used to version the scripts as they changed, and be able to track those changes, but that did not happen. Newer scripts may not have the version defined.

The list of client scripts is sorted roughly in order of appearance during processing

moveTransfer_v0.0

- **Purpose**: Move a Transfer & update database

- **Script**: archivematicaMoveTransfer.py (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/clientScripts/archivematicaMoveTransfer.py)
- **Used in**: Transfer
- **Task type**: once
- **Event?**: No
- **FPR?**: No

Moves the whole Transfer and updates the database with the new location relative to the shared directory.

assignFileUUIDs_v0.0

- **Purpose**: Starts tracking files new to Archivematica
- **Script**: archivematicaAssignFileUUID.py (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/clientScripts/archivematicaAssignFileUUID.py)
- **Used in**: Transfer, Ingest
- **Task type**: per file
- **Event?**: ingestion/reingestion, possibly registration
- **FPR?**: No

This creates an entry in the Files table with the file's UUID, current & original paths and file group. It also creates an 'ingestion' Event and an 'registration' Event if an accession ID was specified. Updating the file group (eg original, preservation, submission documentation) can be disabled with `--disable-update-filegrpuse`.

In ingest, is used on manually normalized files which may have been newly added, metadata and submission documentation.

On reingest, it parses the METS file instead of generating the file UUID, path & group. The Event type is 'reingestion'.

updateSizeAndChecksum_v0.0

- **Purpose**: Set file's size & checksum
- **Script**: archivematicaUpdateSizeAndChecksum.py (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/clientScripts/archivematicaUpdateSizeAndChecksum.py)
- **Used in**: Transfer, Ingest
- **Task type**: per file
- **Event?**: message digest calculation
- **FPR?**: No

Updates the entry in the Files table with a size and checksum. IT also generates a 'message digest calculation' Event.

On reingest, it parses the METS file instead of generating the checksums & sizes. It also re-adds Derivation & Format links.

Note this script will fail if there was a problem with #assignFileUUIDs_v0.0.

createMETS_v0.0

- **Purpose**: Generate the transfer METS file
- **Script**: archivematicaCreateMETS.py (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/clientScripts/archivematicaCreateMETS.py)
- **Used in**: Transfer
- **Task type**: once
- **Event?**: No
- **FPR?**: No

Creates the Transfer METS file. This will contain all the information generated on the transfer during processing, and is especially useful for backlogged transfers.

Not to be confused with #createMETS_v2.0 for the AIP METS.

Improvement note: The Transfer METS file & related backlog functionality needs to be expanded. See Transfer_backlog_requirements#Proposed_improvements for details.

createEvent_v0.0

- **Purpose**: Create an Event outside of other scripts
- **Script**: createEvent.py (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/clientScripts/createEvent.py)
- **Used in**: Transfer
- **Task type**: per file
- **Event?**: as parameter: quarantine, unquarantine, placement in backlog, removal from backlog
- **FPR?**: No

Used to generate events for when a file is placed in or removed from quarantine or backlog. These do not have scripts associated with them, so the event creation is handled here.

archivematicaClamscan_v0.0

- **Purpose**: Check for viruses in incoming files
- **Script**: archivematicaClamscan.py (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/clientScripts/archivematicaClamscan.py)
- **Used in**: Transfer, Ingest
- **Task type**: per file
- **Event?**: virus check
- **FPR?**: No

Runs clamscan on the file and generates a 'virus scan' event. If a scan has been run, it is not run again on the same file.

This is run on incoming files, files after extraction, metadata files and submission documentation. It is not run on normalized files.

sanitizeObjectNames_v0.0

- **Purpose**: Strip problematic characters from filenames
- **Script**: sanitizeObjectNames.py (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/clientScripts/sanitizeObjectNames.py)
- **Used in**: Transfer, Ingest
- **Task type**: once
- **Event?**: name cleanup
- **FPR?**: No
- **Tests**: test_sanitize.py (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/tests/test_sanitize.py)

Sanitize object names by replacing anything that isn't an ASCII letter, number, hyphen, underscore, period or parenthesis. This runs on both files and directories, though only files have Events generated. An Event is generated for a file even if only a parent directory is sanitized.

This used code from sanitizeNames.py (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/clientScripts/sanitizeNames.py)

identifyFileFormat_v0.0

- **Purpose**: Identify a file's format
- **Script**: identifyFileFormat.py (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/clientScripts/identifyFileFormat.py)
- **Used in**: Transfer, Ingest
- **Task type**: per file
- **Event?**: format identification
- **FPR?**: IDCommand & IDRule

One of the most important scripts in Archivematica. Since the file format is used to determine many later actions (extraction, characterization, normalization etc), if this fails many important command later will also fail. This is the only script that uses the FPR that doesn't use the file format as a key for looking up what command to run. Instead, an IDCommand is selected and the output is matched to an IDRule to find the FormatVersion.

There is a short circuit handling of PRONOM ID (PUID) outputs. Since many FormatVersions have PUIDs, and both FIDO & Siegfried output PUIDs, this script looks for a FormatVersion with a given PUID. This reduces the number of IDRules that have to be created.

This also populates the legacy but still required FilesIDs table.

> Improvement Note: Only one identification tool can be run at a time currently. It would be better to allow a cascading of tools. E.g. if a file is identified as a video to subsequently run a tool specialized in identifying different types of video. Similarly, if the default tool failed, we could run a backup tool for a second opinion.

extractContents_v0.0

- **Purpose**: Extract files from packages
- **Script**: extractContents.py (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/clientScripts/extractContents.py)
- **Used in**: Transfer
- **Task type**: once
- **Event?**: unpacking, registration, deletion, message digest calculation
- **FPR?**: extract

Extracts files from a package (e.g. zip, tar, 7z etc) and optionally deletes the package. Generates Events for the new files.

characterizeFile_v0.0

- **Purpose**: Collects characterization information on a file
- **Script**: characterizeFile.py (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/clientScripts/characterizeFile.py)
- **Used in**: Transfer, Ingest
- **Task type**: per file

- **Event?**: No
- **FPR?**: characterization, default_characterization

Collects characterization commands for the provided file, then either

1. Inserts the tool's XML output into the database, or
2. Prints the tool's stdout, for tools which do not output XML

If a tool has no defined characterization commands, then the default will be run instead (currently FITS). Can run multiple characterization commands and log the output of all of them.

validateFile_v1.0

- **Purpose**: Validate a file
- **Script**: validateFile.py (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/clientScripts/validateFile.py)
- **Used in**: Transfer
- **Task type**: per file
- **Event?**: validation
- **FPR?**: validation, default_validation

Validates files are correct, where correctness is defined by the file format.

examineContents_v0.0

- **Purpose**: Run bulk extractor for a detailed analysis of contents
- **Script**: examineContents.py (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/clientScripts/examineContents.py)
- **Used in**: Transfer
- **Task type**: per file
- **Event?**: No
- **FPR?**: No

Runs bulk extractor and stores the outputs in the logs directory.

elasticSearchIndex_v0.0

- **Purpose**: Index the Transfer METS into ElasticSearch when sending files to backlog
- **Script**: elasticSearchIndexProcessTransfer.py (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/clientScripts/elasticSearchIndexProcessTransfer.py)
- **Used in**: Transfer
- **Task type**: once
- **Event?**: No
- **FPR?**: No

The data in ElasticSearch is used by the Backlog tab, SIP Arrangement and the Appraisal tab when dealing with files from backlog. Note that this is not run if the transfer is not sent to backlog (since AM 1.5).

> Improvement note: The client config 'disableElasticsearchIndexing' can disable indexing, but this should be removed, since searching for files in backlog is required functionality.

moveSIP_v0.0

- **Purpose**: Move a SIP & update database
- **Script**: archivematicaMoveSIP.py (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/clientScripts/archivematicaMoveSIP.py)
- **Used in**: Ingest
- **Task type**: once
- **Event?**: No
- **FPR?**: No

Moves the whole SIP and updates the database with the new location relative to the shared directory.

normalize_v1.0

- **Purpose**: Generate a preservation or access derivative
- **Script**: normalize.py (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/clientScripts/normalize.py)
- **Used in**: Ingest
- **Task type**: per file
- **Event?**: normalization
- **FPR?**: Rules: access, preservation, thumbnail, default_access, default_thumbnail. Commands: normalization, verification

One of the most important scripts in Archivematica. This generates the preservation and access derivatives, as well as thumbnails. If manual normalization happens, this instead finds the manually normalized files and links them as derivatives.

The same script is used for generating preservation, access or thumbnail derivatives, controlled by the parameter passed in. In general, normalization is only performed on original files, but may be done on service files. This reads the FPR for the specific command, and may fall back to a default access or thumbnail command.

The output is used to generate the normalization report about normalization attempted and success/failure rates.

For historical reasons, much of the functionality is also implemented in transcoder.py (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/clientScripts/transcoder.py) . It runs an event detail & verification command associated with the normalization command.

On reingest, a 'deletion' Event is created for replaced derivatives.

transcribeFile_v0.0

- **Purpose**: Generate OCR of files.
- **Script**: archivematicaTranscribeFile.py (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/clientScripts/archivematicaTranscribeFile.py)
- **Used in**: Ingest
- **Task type**: per file
- **Event?**: transcription
- **FPR?**: transcription

Optionally generates an OCR file for original files based on FPR entries for transcription. If the original file has no transcription rules, runs on the derivative. The new file is a derivation of the original, has a group of 'text/ocr' and is updated with a UUID, checksum, size etc.

createMETS_v2.0

- **Purpose**: Generate the AIP METS file
- **Script**: archivematicaCreateMETS2.py (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/clientScripts/archivematicaCreateMETS2.py)
- **Used in**: SIP
- **Task type**: once
- **Event?**: No
- **FPR?**: No
- **Tests**:
  - test_create_aip_mets.py (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/tests/test_create_aip_mets.py)
  - test_reingest_mets.py (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/tests/test_reingest_mets.py)

Perhaps the most important script in Archivematica: it creates the AIP METS which contains all the archival metadata generated by previous client scripts.

This script imports from several other files for additional functionality: archivematicaCreateMETSMetadataCSV (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/clientScripts/archivematicaCreateMETSMetadataCSV.py) archivematicaCreateMETSRights (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/clientScripts/archivematicaCreateMETSRights.py) archivematicaCreateMETSRightsDspaceMDRef (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/clientScripts/archivematicaCreateMETSRightsDspaceMDRef.py) archivematicaCreateMETSTrim (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/clientScripts/archivematicaCreateMETSTrim.py)

On reingest, it short-circuits and runs archivematicaCreateMETSReingest (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/clientScripts/archivematicaCreateMETSReingest.py) to update the METS file instead.

Not to be confused with #createMETS_v0.0 for the transfer METS.

storeAIP_v0.0

- **Purpose**: Send the completed AIP to the storage service
- **Script**: storeAIP.py (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/clientScripts/storeAIP.py)
- **Used in**: SIP
- **Task type**: once
- **Event?**: No
- **FPR?**: No

Sends the currently processing AIP to the storage service. The Location is selected from the list of AIP Storage Locations associated with the Pipeline in previous tasks.

- **Purpose**:

- **Script**: [1] (https://github.com/artefactual/archivematica/blob/qa/1.x/src/MCPClient/lib/clientScripts/)
- **Used in**:
- **Task type**:
- **Event?**:
- **FPR?**:
- **Tests**:

Config File

Several config settings are read from `/etc/archivematica/MCPClient/clientConfig.conf` on startup.

Variables in the MCPClient section:

Variable	Description	Default value
MCPArchivematicaServer	URL of the MCP gearman server. Must match the server config file.	localhost:4730
sharedDirectoryMounted	Directory structure owned by Archivematica and shared between the MCPServer & MCPClient. Must match the server config file.	/var/archivematica/sharedDirectory/
archivematicaClientModules	Path to the list of jobs to register with Gearman	/usr/lib/archivematica/MCPClient/archivematicaClientModules
clientScriptsDirectory	Path to the directory where client scripts are installed. Used when parsing archivematicaClientModules	/usr/lib/archivematica/MCPClient/clientScripts/
LoadSupportedCommandsSpecial	Whether or not to register the SupportedCommandsSpecial section of archivematicaClientModules. This should be removed.	True
numberOfTasks	Number of MCPClient workers to created. 0 detects the number of cores and uses that.	0
elasticsearchServer	URL of the ElasticSearch server.	localhost:9200
disableElasticsearchIndexing	If true, do not index AIPs or Transfers in backlog. This should be removed, since ElasticSearch indexing is required	False
temp_dir	Path to the temporary usage directory. Should be in the shared directory	/var/archivematica/sharedDirectory/tmp
kioskMode	Dashboard setting that disables editing users. This should be removed, or at least moved to dashboard settings	False
removableFiles	List of filenames that are not archivally significant and can be removed.	Thumbs.db, Icon, Icon\r, .DS_Store
django_settings_module	Name of the Django settings module, so the client scripts can access the database via the Django ORM.	settings.common

Retrieved from "https://wiki.archivematica.org/index.php?title=MCPClient&oldid=11753"
Category: Development documentation