CatDV Worker Node 7.0

Release Notes and User Manual

Square Box Systems Ltd
Lake View House,
Tournament Fields,
Warwick,
CV34 6RG
United Kingdom

Release Notes

Thank you for downloading the CatDV Worker Node software, the advanced workflow automation and transcoding tool of the CatDV product family.

Important: Please note that it is possible to configure the worker node to automatically delete files after they have been processed (as this may form a valid part of the workflow). Take great care therefore when configuring such workflows: make sure the root of the watch folder is set correctly and test all your scripts carefully before relying on them for critical data!

Licensing

To use this software you will need a valid Worker Node 7.0 license code, either a worker license issued since 1 May 2017 or be under a current software maintenance agreement.

CatDV and the Worker Node are regularly updated with both fixes and functional enhancements. If you are under a current software maintenance agreement you can download all updates, include major version upgrades, as long as your agreement is up to date. If you are not under maintenance you can't use major version upgrades and can download minor bug fix point updates for a period of six months from when your license was issued only. Please contact sales@squarebox.com if you need an evaluation license or have any other queries.

Worker Node licenses are available in different editions:

- Restricted Worker – only available bundled as part of certain archive solutions without the ability to configure general purpose watch actions
- Workgroup Worker – entry level system supporting up to 3 automated watch actions
- Enterprise Worker – fully featured core system supporting an unlimited number of watch actions and recommended for most users
- Pegasus Worker – high end system designed for multi-node and multi-site operation, including support for remote monitoring and worker farming

Beta expiry

This is a public beta and will expire on 10 February 2018.

Upgrading from Worker 6

Please note that if you are upgrading from Worker 6 you should back up your old worker config and task list files as once you use Worker 7 these files are not backwardly compatible with previous versions. You should test your workflow thoroughly before relying on it in a live system. Also be aware that switching to the new more efficient task list file format will involve clearing out existing tasks.

Requirements

The requirements to run the worker node software are very similar to those for running the CatDV client application. Please consult the CatDV release notes for further details. Under Mac OS X no additional software is needed, under Windows you may optionally install QuickTime (http://www.apple.com/quicktime/download) to provide access to additional codecs.

The worker is designed for use with CatDV Server 7.2.0 or later.

New Features in Worker 7.0

Worker 7.0 is a major update to the worker with the following main changes:

- Worker 7.0 is built on the CatDV 12 code base and includes improvements to the core engine in CatDV 12, including better metadata extraction (from MPEG, JPEG, WAV and MOV files), improvements to the FFmpeg exporter, support for automatically creating a .jpg proxy when building proxies from a still image, ability to use syntax like clip.name or media['FNumber'] in variable expressions, support for Sony XAVC-S and Canon XF-AVC metaclips, improved AAF and FCPXML support, ability to create audio waveform thumbnails from audio files, and support for XMLv2 sidecar files.
- Like CatDV 12, Worker 7.0 features a more modern flatter look and feel, and native support for the CatDV Server 7 metadata schema.
- Worker 7.0 is able to run without QuickTime being installed. If QuickTime is available it will be used, as it provides access to Apple ProRes and third party codecs such as from Avid and Calibrated, but if it isn't installed the Worker can still process most files using its build in media file analysis and metadata extraction routines and transcoding capabilities provided by FFmpeg.
- As a result of reducing the worker's dependency on QuickTime we are now able to offer a version of the worker that can run under various versions of Linux (to enable deployment on the cloud for example).
- The worker engine now runs as a separate process from the user interface. There are new buttons to explicitly start and stop the worker running. Editing the configuration no longer has to immediately apply the changes and restart the worker and it is possible to quit the worker user interface (i.e. the config editor and status/task list monitoring application) while leaving the core worker engine itself running as a background process.
- The worker can be installed as a background service that is started automatically when the system boots up, rather than having to be started manually in a logged in user session.
- There is the option to store the task list using an embedded database rather than using the old xml file based format as this is more efficient and permits faster throughput.
- There is a new graphical configuration editor that displays a visual overview of all the watch definitions and processing steps. Different colours indicate the type of processing step, and "stacked" nodes indicate where a single watch action such as a server query can queue multiple separate tasks. Double click a node to edit the watch definition, then drag and drop available processing steps across to edit the watch action.
- The worker main window includes a new task summary panel that shows how many tasks completed or failed, without needing to keep the task list panel open to to monitor failed failed tasks. Additionally, it is possible to have multiple task list panels open with different task filters, and to undock or remove the task list from the main window.
- The worker supports a new Execute Javascript processing step, and also supports plugin extensions (for example a File Path Cleaner or Upload To YouTube processing step) that can be purchased from our market place.

- It is possible to disable watch actions so they don't run automatically, but instead run them manually from the Run Now menu, or with the new Run Watch Action processing step. This allows one watch action to trigger another watch action immediately without requiring polling.
- Better control over task priorities (lowest priority watch actions will only run and queue new tasks when the system is idle, and if you have three or more worker processes then at least one process is reserved for normal and high priority tasks only).

New Features in Worker 6.1 and earlier

Worker 6.1 is a major update to Worker 6.0 that:

- Increases the use of processing steps, by replacing the final copy/move/delete file step and the setting of clip fields on the Publish tab with processing steps, adding more flexibility and allowing the user interface to be simplified.
- Adds support for conditional processing by making each processing step conditional. Only those clips which meet a condition (those where a specified field matches a regular expression pattern) will be processed by that step, and if no clips match then that step will be skipped entirely. Taken together with server query triggered tasks and the Test File Existence processing step this allows a great deal of flexibility when developing worker scripts.
- Adds a new option to resume the worker after a few minutes if a server error occurs (normally unexpected server errors cause the worker to pause, allowing you time to resolve the problem manually, but you may want to select this option for unattended processing).
- Adds support for limiting the number of files that will be processed at a time in one quick folder scan or bulk server query step.
- Adds support for camera card manifests to ensure that a complete camera card is copied or archived when dealing with complex clips (P2, XDCAM, XF, XAVC, AVCHD, etc.). Additionally, the media path for metaclips is shortened and made more consistent so it doesn't include unnecessary path components like BPAV/CLPR or CONTENTS.
- Allows array syntax to pull out parts of a file name in variable expressions.
- Supports rotation of worker log files (by default a new log file is created once per day or when the log file gets over 40MB in size, or when you reinitialise the worker, and the five most recent files are kept).
- Add support for importing and transcoding DPX and EXR image sequences.
- Add support for transcoding from any supported media format to any other via intermediate image sequences (for example, from RED or AVCHD directly to QuickTime ProRes).
- Improvements to file preprocessing, with explicit control over whether to copy then delete files to move them across file systems, and options on how to deal with error conditions. (Use the copy/move directory command to cope with complex media types and the regular copy/move file command for normal files.)

Worker 6.0 adds the following main features:

- The main change architecturally is that Worker 6 has been updated to work the same way as CatDV 11. The application now runs 64-bit for improved performance and memory handling and uses the new native helper process for import and transcoding of files (instead of the older

QuickTime for Java technology). It also comes bundled with Java. These changes ensure compatibility with newer system software such as Mac OS X Yosemite.

- Worker 6 provides access to the new Native QT and FFmpeg transcoders of CatDV 11 (replacing the old 32-bit Xuggle, DSJ, and QTJava technologies which are now obsolete)
- There is a new CatDV XML v2 file format for importing and exporting clips which provides full access to the CatDV data model, including sequences, metaclips, event markers, metadata fields, thumbnails, and more. It is possible to update selected fields of existing clips via XML, as well as inserting new clips (both online and offline), and to automatically delete existing clips if required. The old CatDV XML batch file format continues to be supported.
- The 'Conversions' tab has a new panel where any number of processing steps can be added, replacing the fixed command execution and XML export fields. This provides a lot of additional flexibility and allows related files to be copied or moved, multiple commands to be executed, etc. Movie Conversions (proxy generation, and exporting stills and standalone movies) use this mechanism too, allowing conversions and other types of processing steps to be intermingled as required. Existing Worker 5 scripts are automatically converted to use the new processing steps (and a backup copy of the old worker script is kept).
- Similarly, there is a new batch export panel to export all the clips returned by a server query in different formats including AAF, ALE, EDL, CDV catalogs, and FCP7, FCPX or CatDV XML formats, and also to execute an external command on the exported clips.
- Improvements to Quick Folder Scan and the ability to test whether files original media or proxy files exist make it easier to keep catalogs and disk folders in sync and perform conditional actions if certain files are present or missing.
- There is a new 'scheduled job' trigger type for performing actions such as a database backup at specified times. The user interface for defining watch actions has been updated to expand the description of different trigger types.
- Metadata extraction rules can be applied to automatically populate metadata fields based on file naming conventions when importing a media file.
- The task list viewer in the main window has various options to filter the tasks based on their status or by searching for text in the file path or task summary.
- It is possible to hide watch actions which are no longer being used but which you don't want to delete yet.
- 'Allow polling' has been changed to 'remember completed tasks' and the sense of this flag has been inverted. Wherever possible you should write watch actions so they remove the trigger condition once the task has been completed (change the status of a clip so it isn't returned by the server query again, move a file out of the watch folder once it has been processed, etc.) and you don't rely on the worker remembering completed tasks as this is much more efficient.
- It is possible to import and export individual watch actions and conversion presets, making it easier for systems integrators to provide and install customised scripts for particular tasks. These are saved as an XML file with extension .catdv which can be dragged into the worker node window to install it.

- There is a new job description field for adding optional notes about what a watch action does, and an optional worker name field to help identify which worker is which if you have multiple instances.
- There is a new field chooser when building server queries and setting metadata fields which allows you to type in the id of an arbitrary “blue” metadata field that might not appear in the list, eg. @ExifVersion.
- There is a new wizard to help you configure the destination path for a file export or move operation, including a regular expression previewer that will evaluate and display the value of an expression as you make changes.
- There is a new wizard to help with configuring command lines, including the ability to pick files for each individual argument and built-in templates for common actions such as Episode transcoding or MySQL backups.
- There is a new ‘Dump Variables’ processing step to help with debugging worker scripts.

CatDV Worker 5.1 builds on Worker 5.0 and provides additional flexibilty in terms of the types of work action that are supported, in particular for dealing with large numbers of clips in one operation:

- If you don’t need to perform slow operations like transcoding or thumbnail extraction on each file it is now possible to do batch operations such as a bulk server query and update or importing an entire folder of files in one operation
- A new ‘re-analyse media’ action allows basic stub clips to be created in the database first and then the detailed media file analysis and thumbnail extraction to be performed later (either once the file is available, or based on a server query)
- Ability to batch import clips from a CatDV XML Batch File even if the media files are not available, providing an easy way for external systems to insert clips into the database
- Other new features such as RED metaclips, automatic calculation of MD5 checksums, support for preserving P2 and XDCAM folder structures when copying and moving files, and improved handling of Unicode characters have also been added since Worker 5.0. See the Release History below for a detailed list of changes.

The following features were added in CatDV Worker 5.0:

- New framework for working with additional media libraries (such as JMF or Xuggle, rather than just QuickTime) for both importing and transcoding media files
- Improved QuickTime exporter and rewritten settings dialog, including ability to set QuickTime annotations, burn in text using complex variable expressions, override the reel name when adding a QT timecode track, better control over specifying the frame size, and set the background colour when burning in text or timecode.
- New Xuggle exporter, giving access to additional import and export formats, including the ability to deinterlace the video by line doubling and burn in text or timecode
- New DSJ exporter (Windows only) giving access to Windows Media and DirectShow codecs. (Note: on Windows 2008 Server it may be necessary to enable the “Desktop Experience” feature via Manage Computer to provide access to the DirectShow/Windows Media libraries.)
- Update the processing engine from CatDV 8 to CatDV 9/10 code base, so that changes made since CatDV 8.1.12 are now included (and ensure they continue to stay up to date in future).

This includes changes such as being able to select the Xuggle or JMF importers (if installed), support for reading IPTC metadata, and improved metaclip support.

- Ability to run the task processor in a separate process, protecting against the possibility of crashes on particular media files. (Note that running in a separate process is not supported under Windows XP and requires Vista or later.)
- Support for exporting all the clips that match a server query to HTML or XML as a single batch, rather than separately for each clip.
- Simplify the worker by removing the ability to define more than one work set.
- Instead, allow for multiple worker processes to run concurrently for increased throughput (note that each worker will require its own license). Tasks can be assigned a timeout value and will be terminated after that time or if the external processor stops responding.
- Allow scheduling of tasks so they only run on certain days of the week or certain times of the day. This affects both when new jobs are created (via a watch folder or by running a server query) and when previously queued jobs are permitted to run.
- Incorporate the task list in the main window
- Remove the "pending" task state. Clips are published to the server immediately rather than being batched up. Additionally, appending to an existing catalog is more efficient as the worker no longer opens the catalog by loading existing clips from the server simply in order to append to the catalog.
- More logical arrangement of the user interface for defining a watch action, with changes to the post-processing and trigger tabs
- Increased number of extra fields that can be set on a clip
- The Windows version now comes with an .exe installer
- Ability to restrict watch folder actions to known file types such as common media files or P2 folder structures, based on the file name and extension
- New development mode to speed up testing by reducing poll and settle periods to 10s and prompting for confirmation before starting each task
- New "Slate" look and feel (this can be turned off if necessary on the General config page)
- The worker.xml file and default workset location (though you can change the latter if required) have moved from your home directory to the application data directory. After migrating to Worker 5 you can remove the old worker files to avoid cluttering your home directory.

These are in addition to the following features which were added in Worker 4.0:

- Support for unlimited server query terms
- Improved management of watch definitions, including a new 'test' button for watch folders and server queries and the ability to 'solo' individual actions by shift-clicking the Enabled button
- Support for media path mapping
- Improved handling of jobs submitted via XML batch files

- New options to use time of day timecode, add watermarks, and perform two-pass transcoding to add a second audio track with different settings
- Support for loading user-defined field names from the server
- Ability to specify separate emails for success and failure notifications
- Ability to perform processing on completion of an xml or server triggered batch
- Support for regular expressions to modify variable values
- A major new command line interface to generate reports or perform updates on the database for use within external scripts.

Release History

7.0b19 8 Dec 2017

JavaScript and plugin enhancements and fixes. Fixes related to quick folder scan handling of camera cards. Fix an issue with watch folders if the settle period is set too high. Fix an issue with intermediate publish steps on a bulk query action. Fix an issue with accessing worker variables. Fixes to the graphical editor. Improved prioritisation of high priority jobs. New segment-based FFmpeg exporter that can burn in markers and a title slate. Fix cleanup of intermediate transcode files. Update CatDV engine to match CatDV 12.1.0.

7.0b18 17 Oct 2017

Add support for entering multiple license codes and running licensed worker extensions. Bulk queries can efficiently move clips to another catalog based on a clip variable expression. Lowest priority watch actions only run the query or scan the watch folder when the task queue is empty. New processing step to manually trigger another watch action ("run now") to allow watch actions to be chained. If you have four or more processes the last two are reserved for normal and high priority tasks so low priority tasks will never block a high priority one (if have 3 processes the last process is reserved). The task details panel has a button to take you straight to the appropriate section of the worker log. JavaScript enhancements. Fix related to path mapping when applied to metaclips. Various other fixes.

7.0a17 3 Oct 2017

Fixes to drag and drop in the graphical editor. Fix an issue where too many licenses were used up. Fix an issue with the test email button. Add filters to the graphical editor to make it easy to manage lots of watch actions. Change Execute JavaScript step so it doesn't expand variables inline but instead provide an explicit expand() function. Simplify licensing by removing the need to allocate processes when entering licenses (so a 4-session license, for example, is always valid for one worker installation running four parallel processes).

7.0a16 13 Sept 2017

Fix an issue with watch folders with trailing file separator. 'Update existing' will replace any markers of the same category as it finds in the file. Fix an issue with configuring conditional steps. Server 7 'All Fields' field set is hardcoded unless legacy option is selected. Fix ability to set task priority from the task list. Fix importing of certain .mxf, .ts and .mov files. Fixes to AVB importer. The 'set variable' processing step now allows longer variable names like ${xyz} to be used. Fix an issue where the user interface wouldn't launch on certain Windows machines.

7.0a15 4 August 2017

New "AVB Folder Sync" importer. Minor fixes related to custom catalog fields, bulk queries with a query limit, regex tester with Windows file paths.

7.0a14 1 August 2017

Improved reliability connecting to worker service. JavaScript improvements. Set Clip Fields has a value chooser button. Make sure field names are available in config editor. Other fixes and improvements.

7.0a13 5 July 2017

New processing step to reanalyse media files on demand. Improvements and fixes to graphical editor. Improvements to JavaScript support. Improvements to file chooser. Other fixes related to moving sidecar files, transcoding audio, path mapping, and more.

7.0a12 18 June 2017

New Execute Javascript processing step. Fixes and improvements to the graphical editor. Add a 'Run Now' menu command to manually run watch actions, including actions which are disabled from running automatically. Add a 'Refresh' to manually refresh the task list if required (though it should normally update automatically). Add support for plug-in extensions. Fix viewing job xml. XMLv2 now supports relative timecode on markers. Worker command line on Windows is now 64-bit. Fixes to importing image sequences via the worker. Other fixes.

7.0a11 18 May 2017

Improvements to graphical editor. Fix an issue where tasks on hold could cause the same task to be queued multiple times. Fix "File is null" issue when editing config. Add support for transcoding slide show sequences using FFmpeg. Fix issue with query on metadata fields. Allow creation of mezzanine clip from sequences by specifying an explicit path using variables in the mezzanine root path field. Other fixes.

7.0a10 5 April 2017

Allow engine to be started and stopped while the service keeps running. Allow the service to be installed as a background service (Mac only, experimental only, at this stage). Remove a dependency on having a legacy Java 6 installation under MacOS. Variable expressions can be written with array notation after the regex, eg. $i{s/mov/mp4}[0]. Improvements to the graphical editor. Fix development mode. Add support for reading XAVC-S card volumes.

7.0a9 20 March 2017

Fix an issue with timeouts when starting the worker service. Add a new task summary panel to the main window. Add the ability to detach the task list from the main window and to open additional task list panels (with different task filters) from the View menu. Add the ability to turn off the button panel and use the menu instead to start and stop the service, edit the worker config, etc. Add an option to the general config tab to switch to storing the task list in a database rather than in a workset xml file. Update the CatDV engine to 12.1, including updating the bundled version of java to 1.8 and the version of ffmpeg to 3.2.

7.0a8 16 February 2017

Fix an issue where the task list might stop updating.

7.0a7 8 February 2017

Improved robustness if another worker instance is running, if there are errors when starting the worker service, etc. Don't terminate the file watcher if there is an error processing an xml batch file.

7.0a6 30 January 2017

Extend the graphical editor so it can be used to create new watch actions and support drag and drop editing of existing watch actions. Changes to the way the workset is persisted.

7.0a5 13 January 2017

Separate the worker engine out so it runs as a separate background process rather than as part of the main user interface application. Add a graphical overview panel to the configuration editor. Add support for filtering the task list by job name.

7.0a4 31 December 2016

Add buttons to explicitly start and stop the worker process and change it so that editing the configuration doesn't have to immediately apply the changes and restart the worker. Minor fixes to the UI and when running under Linux.

7.0a3 26 December 2016

Updated worker engine to match CatDV 12.0.2

7.0a2 31 October 2016

Fix Linux issue with missing file.

7.0a1 7 October 2016

Initial alpha release with support for Linux

6.1.9 14 November 2016

Fix an issue with loading user field definition set from set that was saved using CatDV 12. FFmpeg exporter has option to create a .jpg proxy for still images (rather than a single frame .mp4 etc). Bulk queries now let you use clip variables in processing steps (except for Set Clip Field and Test File Existence, which implicitly iterate over all the clips in turn, the first clip returned by the query will be used). Fix an issue with the command line -distinct option.

6.1.8 21 October 2016

Fix an issue when reanalysing non media files. Various FFmpeg native helper fixes. Fix command line -summarise option on multigrouping values. Fix an issue when searching/filtering large log files. Fix an issue where transcodes that time out are reported as having been cancelled by the user. Allow access to ARCHV variable. Ensure moving file.mov also moves a sidecar called either file.mov.xml or file.xml. Automatically load latest field definitions from server when entering config editor. New processing step to set worker variables like $X (so you can do things like calculate a temporary filename just once and then refer it to it in subsequent steps).

6.1.7b 8 August 2016

New option when building mezzanine clips to replace the clip being worked on (allowing different metadata to be set on the original and mezzanine clip and allowing the mezzanine clip to be moved to a different catalog). Improvements to "Copy to another catalog" functionality where you can set different metadata on the original and copied clip by using an intermediate Publish To Server processing step to update the original clip; also $a now refers to the new catalog. Allow email batch interval of 0 to send messages immediately with no batching. Fixes related to cron job scheduling. Fixes related to XAVC metaclips.

6.1.6 25 Jun 2016

Fix issues relating to building mezzanine movies (including error handling if destination already exists or source isn't a media file, and "you can't add new clips to query results" messages). Fix handling of AVCHD metaclips. Fix 'reanalyse' function so it can convert a standard clip to a metaclip. Add 'import as AAF' option. Read iXML metadata from WAV files. Better handling of errors such as files with timestamps in the future and failure to write the workset file. Fixes related to moving metaclips (it is now possible to use a 'move file' preprocessing step with camera cards, including with the 'delete empty directories' option and when using ghost folders). Ensure $N variable (based on pseudo path for metaclips) is available early enough to be used for catalog names. Update engine to match CatDV 11.1.7 (including ability to run without QuickTime). Fix an issue with launching the command line. Other fixes and improvements.

6.1.5 13 May 2016

Fix an issue where the media path might not be updated if an intermediate publish step fails. Fix checking whether complex clips are online. Fix an issue where worker might fail to save workset file after updating from an old worker version. Make sure Bin name is set when importing audio files. Fix an issue with the command line when running against server 6.10. Other minor fixes.

6.1.5b 31 Mar 2016

Fixes related to XMLv2 (include missing metadata when exporting metaclips, include <mappedPath> if file mapping is being used and media path doesn't match current location, fix an issue importing if filesize is zero). Add new 'folder clip' importer if clips representing entire folders without scanning contents are required.

6.1.4 4 Mar 2016

Fix an issue where server triggered tasks could sometimes be resubmitted incorrectly. Fix an xml issue with transcode processing steps. Improvements to FFmpeg exporter (new option to preserve all audio streams). Fix an issue when running queries against Server 7 from the command line. Logging improvements. Fix an issue where two intermediate image transcodes could interfere with each other. Improved reporting of audio format for AVCHD and MPEG4 files. New option to read back and do bitwise verification whenever a file is copied. Fix an issue related to the consolidate media action.

6.1.3 13 Feb 2016

Add an option whether to check for files being online or not as this can be very slow on some file systems. Fix an issue with Quick Folder Scan when applied to complex camera cards (resulting in incorrect path or duplicate clips being published). Fix an issue with the task summary not always being updated correctly. Fix an issue updating thumbnails via XMLv2. Fix an issue with Test File Existance step and add a $N variable to help with testing for proxies. Add the ability to send emails as a processing step. Option to copy metadata from original file when transcoding to mezzanine movie. Improvements to the FFmpeg exporter (support for transcoding sequences and metaclips, better selection of x264 quality settings). Variable expressions can use Javascript as an alternative to regular expressions. Update engine to match CatDV 11.1.4. Other fixes and improvements.

6.1.2 15 Dec 2015

Add 'Check existing' option to control whether copying or moving a file checks to see if the exact file already exists at the destination and if so treats it as having been copied (rather than failing or picking a new name). Command line improvements (allow media path to be set even if clip has no media object, new –insertclip clip, add TF field, sign the command line executable on Mac). Add support for DPX and EXR image sequences. Add support for 'Workgroup Worker Node' licenses (restricted to 3 watch actions). Allow 'Restricted Worker

Node' license to edit path mappings. XMLv2 enhancements (allow updating existing thumbnails, new thumbnails replace option, new updateLocal option). New processing steps to perform housekeeping or broadcast a message. Fix an issue where period of scheduled cron jobs might be incorrect. When creating a new conversion preset allow one of the built in templates to be chosen.

6.1.1 3 Nov 2015

Fix an issue with path mapping between Mac and Windows paths. Fix an issue with exporting stills. Add a new 'Consolidate Media' processing step that writes new files to collect the media for those parts of a subclip or sequence that are actually used together in one place, and updates the clips to point to the new files. Update the version of FFmpeg and RED SDK.

6.1.0 5 Oct 2015

Official release of 6.1.

6.1b8 14 Sep 2015

Add support for DPX and EXR image sequences. Add support for transcoding from any supported media format to any other via intermediate image sequences. New preprocessing options to explicitly control whether to move files by copying then deleting the file and what to do if there's an error. Preprocessing copy or move should apply to xml sidecar files also. The copy/move directory commands will now work with files at the root level of the watch folder. When creating a mezzanine clip, set OriginalClipID to refer to the original clip. Improvements to the log file viewer. Other fixes and improvements.

6.1b7 20 Aug 2015

Automatically rotate log files at midnight and when they get too large and keep them together in a separate 'CatDV Worker' log file directory. Improvements to the log file viewer. Add a new processing step to create mezzanine files which transcodes a movie and then reimports the file as a new clip. Fix various issues related to: MD5 calculation, copying of empty folders, deletion of metaclips, deleting empty folders, and having a generic dock icon on Mac OS X.

6.1b6 12 Aug 2015

Add support (as per CatDV 11.1) for linking all the clips that relate to one raw camera card volume using CameraCardID metadata field and create a new camera card manifest metaclip that describes empty directories and card level index files, also create AVCHD metaclips. New option when performing server queries to include all the clips from those cards. Ensure destination path is calculated consistently when copying and moving metaclips. Fix an issue with updating the media path after copying or moving metaclips. Fix an issue where MXF files could be locked after importing them so they couldn't be moved. Add a new regex helper to the Help panel.

6.1b5 21 July 2015

Fix an issue with multiple Publish Changes processing steps in the same action. Add a missing option to FFmpeg exporter to make sure resulting files are playable in QuickTime. Fix MXF importer so it doesn't keep files locked. Add a query limit to server query triggered actions. Add a batch size limit to Quick Folder Scan actions. Support array syntax as a simpler alternative to regular expressions to unpick parts of a file path. Reduce verbosity of the log files. Extend the "don't suspend worker if a server error occurs" option so it applies to server monitor as well as worker threads.

6.1b4 11 July 2015

Increase memory allocation for Windows worker. User latest FFmpeg version. Add $_B (batch id) and $_? (matching filename in wildcard file existance test) variables.

6.1b3 30 June 2015

Fix various issues with new file watcher (ignores the include subdirectories option, doesn't correctly remove duplicates, still has slow performance on certain Windows network shares). Fix an inconsistency where 'reanalyse media' could result in a task failing while importing the file as a new asset would work. Fix reanalysis of metaclips. Fix an issue where the worker always cancels system shutdown on a Mac. Add a workaround for intermittent "update failed" errors. Rename Quick Folder Sync to Quick Folder Scan. Other minor fixes and improvements.

6.1b2 9 June 2015

Add support for conditional execution of individual processing steps. Fix issues related to the regex wizard, with the set clip status processing step, reporting when file deletion fails, and an error extracting thumbnails when no audio hardware is available. Display the process number that was used to run a task, making it easier to find the relevant helper log file when debugging scripts. New option to resume the worker if a server error occurs.

6.1a1 3 June 2015

Use new Java file APIs when scanning a watch folder for improved performance. Combine Publish tab with Processing Steps and use processing steps to set clip variables, save a local catalog, and perform final file copy, move or delete operation. New file type filters on the Conditions tab that check for common still image (jpg, tif, png, psd, cr2, raw, nef) and movie (mov, mp4, avi, wmv, mpg, mts, m2t, mxf) file name extensions.

6.0.1 24 May 2015

Only remember old tasks in the task list if configured to do so. Allow export of multiple watch actions to a single .catdv worker script bundle. Fix an issue related to reordering watch actions. Fix an issue with file contents analysis on the Conditions tab. Fix an issue where the task list would sometimes not refresh. Fixes related to including quotes in arguments when executing commands. New wizard to help when configuring command execution steps, including build in command templates. New 'Dump Variables' processing step. Fix warning message about old worker license not being valid. Fix issue with 'no thumbnails' not working.

6.0.0 28 April 2015

Public release. Include new file path wizard to help configure processing steps and preview effect of regular expressions. New $_i ISO date variable. Other minor fixes.

6.0rc1 22 April 2015

Release candidate. Change Movie Conversions so they are now defined as part of processing steps. Change 'allow polling' to 'remember completed tasks' and invert the sense of the flag as that better describes what it does. Fix an issue where even if you remember completed server query tasks, if you edit a clip back so it's triggered again it should be processed again without having to restart the worker. Other fixes related to 'write text file' and scheduled tasks.

6.0b8 15 April 2015

Quick Folder Scan will now create metaclips. New Test File Existance processing step to check whether media file or proxy is available and update the clip accordingly. New Set Clip Field step and ability to publish to server more than once. Fix an issue with duplicating watch actions. Allow $y (stderr output) to be used in the body of an email. Allow media path to be set to either first or last file to be copied. Other fixes and improvements. New documentation section on syncing a folder with a catalog.

6.0b7 30 March 2015

Expand description of trigger types and add support for scheduled jobs. Allow unused watch actions which you don't want to delete yet to be marked as hidden. (If any watch actions seem to be missing after your update click the 'Show all' button!) Copy/move processing steps should copy all the files in a metaclip. Display better diagnostics if a file copy/move operation fails. Add options to ignore missing source files in copy/move operations, and to ignore exit status when executing a command. Fix missing thumbnails when exporting XML files. Add next/previous button when viewing task details. Retry automatically if publishing to server results in a save conflict. Fix migration of old Worker 5 scripts where certain processing steps could be lost. Prevent command execution from hanging in some situations by consuming console output. Fixes related to importing of CatDV XML v2 files.

6.0b6 27 February 2015

New field chooser with filtering and the ability to query and set arbitrary metadata fields such as @FNumber etc. Improvements to list of processing steps (specify what to do on copy or move if destination file exists, configurable HTML fields, ability to write a text file using variables, etc.). Make it easier to include and exclude patterns on the file conditions tab (if you tick the regex option the pattern is searched anywhere in the path but don't need to exactly match the entire path). You can enter an optional worker name on the General tab which is displayed in the window title to help identify different worker instances. Option to delete clips via XML v2 if required.

6.0b5 17 February 2015

The conversions tab now supports a list of processing steps, allowing any number of commands to be executed and related files to be copied or moved. Existing job definitions are automatically migrated with minor differences in functionality (any command not just the first one will update $x and $y, and exporting XML and running command 2 now happens before the file copy/move step). The file copy/move step under "Options" maintains old functionality and supports metaclips and consecutive file numbering. Allow advanced properties to be specified on a per task basis if required. Allow Final Cut XML to be explicitly chosen as an importer (works with both FCP7 and FCPX files). New option whether to perform media analysis of files referenced by a CatDV or FCP XML file. If an MD5 checksum is calculated this is calculated early on, before any file pre-processing step.

6.0b4 11 February 2015

Changes related to importing XMLv2 files (reading clips with filesize 0, display progress indication while importing, more searchBy options, allow adding clips to multiple catalogs during a single import if the catalog name being published to contains variables, including ${CAT} to take the catalog name from the xml file). Fix an issue with the Automatic importer not correctly identifying which importer to use. When exporting XMLv2 files allow an <options> directive to be inserted automatically in the xml file. New 'CatDV Catalog (Bulk)' exporter which exports entire catalog if the server query returns any clips in that catalog (combined with the CatDV Catalog importer and the 'replace catalogs' option this allows catalogs to be sync'd from a master to a slave server). New 'since last scan' option when performing batch queries.

6.0b3 8 February 2015

Add new CatDV Catalog Importer with the option to replace an existing catalog on the server in its entirety. Folder Sync (Quick) importer can now put clips in different catalogs based on the file name. XML v2 now has ability to merge individual metadata fields. New mechanism to allow import processing rules to work between Mac and PC. Other fixes.

6.0a2 22 January 2015

Fix issues running under Windows. Add mechanism to parameterise .catdv settings files. Add $? variable for last command exit status.

6.0a1 13 January 2015

Initial release using CatDV 11 codebase. Obsolete QTJava, DirectShow, Xuggle and JMF technologies have been removed. Other changes include task filtering, XML version 2, batch export actions, metadata extraction rules, and the ability to export watch action and conversion presets.

5.1.3 1 December 2014

Allow import of .cdv catalog files. Support XML sidecar files when importing TIFF images. Improved support for archive integrations (ability to set ARCHV field and new restricted license type to support dedicated integrations). Allow Java –Xmx options to be passed in when invoking worker command line tool. Update engine to match CatDV 10.1.3.

5.1.2 26 August 2014

Performance improvements when opening large catalogs and doing quick folder sync. Support for clips whose timecode wraps at midnight. Fix out of memory error that can occur with certain malformed variable expressions. Improvements to behaviour of watch action 'Test' button. Update engine to match CatDV 10.1.2.

5.1.1 29 June 2014

Fix an issue when renaming files using a variable expression. Fix handling of spanned Canon XF clips. Fix aspext ratio of CR2 raw images. Provide timecode as a time in seconds or as frames using :t or :f format modifiers in variable expression. New option when exporting standalone movie to use the In2/Out2 selection. Respect FCP7 field mappings in the field definition set. Update the version of java mail library. Increase time out when launching helper process. Ensure server query 'allow polling' option will keep processing the same clip. Fix an issue when multiple worker processes detect the same spanned P2 clips. Allow "Move folder" to copy then delete files across file system boundaries (requires mvdir.copydel=true option).

5.1.0 3 Apr 2014

Public release of Worker 5.1. Exporting stills will revert to exporting thumbnail if movie is unavailable. Enhance batch file import so it works with path mapping. Fix -43 error when exporting movies containing Unicode characters on Windows. Changes to support export of TIFF or PNG stills using the QuickTime image sequence exporter.

5.1b1 (5.0.9b) 9 Mar 2014

Ability to perform bulk server-triggered tasks (see new documentation section on ‘Individual vs batch actions’ below). Change default character encoding to UTF8 and add new option to log viewer. Command line fix on Windows.

5.0.8 15 Feb 2014

Fix related to determining which task to run next based on priority. Update engine to match CatDV 10.0.9 (fix import of certain XDCAM files, fix -1409 error when exporting stills, read additional Avid MXF metadata for use with CatDV Pegasus). Allow use of variables from CatDV XML batch files when naming catalog to publish to, e.g. `${@CATALOG}`. Add option to Avoid Duplicates in the destination catalog when copying or moving clips from a server-triggered task to another catalog. Minor user interface improvements (related to scheduling, batch file processing, and handling of P2 folder structures).

5.0.7 19 Nov 2013

Fix issue with path mappings on server queries not working. Implement pick list chooser (Values button) when editing server queries. New “CatDV XML Batch File (Quick)” importer which quickly imports clips without doing a full media analsysis (ie. as if the files were offline). New “Folder Sync” importer which keeps a catalog in sync with a watch folder by doing a quick import of files without doing a full media analsysis, including new $K and $L variables. New option to calculate MD5 checksums on import. Allow import of XML files created by the worker command line. Allow accented and other Unicode characters in emails. Change name of command line on Windows to catdv.exe, so it matches what the documentation says. Other fixes.

5.0.6 11 Sept 2013

Add support for RED metaclips. Replace “Merge MXF files into a metaclip” checkbox with three radio button options. Improve detection of metaclip folder structures on the file watch “Conditions” tab. Moving or copying a metaclip (using the Conversion, not the Pre-Processing tab) preserves folder structure. Improve deletion of empty directories. “Ignore errors” option now applies to movie conversion (as well as file copying and command execution). New -distinct and tabNoHeader options in command line interface. Fix related to resubmitting jobs. Bring engine up to date with CatDV 10.0.7 (fix import of .mts files, read JFIF metadata, FCP XML improvements, etc.).

5.0.5 12 July 2013

Fix an error with server-triggered tasks if reanalyse media option is turned off.

5.0.4 11 July 2013

Fix an issue with the new ‘CatDV XML File’ importer option. Add support for reanalysing media when running a server query-triggered action. New option whether to interpret contents of batch and log files when using the Automatic importer. Add support for reading metadata from RED files. Updated “Importing media files” and added new “Offline media” documentation sections.

5.0.3 6 June 2013

Fix Mac/Windows path error when copying metaclips. New CatDV Batch XML importer to import a batch file immediately, rather than queuing each file, including support for creating offline clips with no corresponding media file. New option to update any existing clip with same path or clip name, allowing clips to be inserted into database prior to the media arriving (similar to CatDV's Reanalyse Media command). Exporting stills now correctly takes into account JPEG quality setting. Update engine to match CatDV 10.0.4 (fixes and improvements related to metaclips, XML sidecar files, MXF importer, IPTC metadata support, etc.)

5.0.2 24 February 2013

Fix "timeout" and "unexpected response" errors when communicating with helper process. New option to delete empty directories after deleting or moving files out of the watch folder. Ensure worker is suspended if there is an error communicating with the CatDV Server. Fix unresponsive GUI when worker is processing a task. Allow files to be moved across file systems. Email recipients can now contain variables. Update engine to match CatDV 10.0.2 (support for raw image metaclips, new Pulldown metadata field, fix -2015 errors, etc.). Better support for importing metadata from XML sidecar files. Other minor fixes. Added a "Hints and tips" section to the documentation below.

5.0.1 7 December 2012

Update engine to match CatDV 10.0.1 (including support for spanned Canon XF clips, improved TIFF support, and prevent thumbnails over 64K). Ensure media path is updated when publishing changes if you use a path mapping. Fix handling of thumbnail and scene detection options. New option whether task should continue or not if there is an error copying the file or executing an external command. Log viewer now supports Cmd-F to search.

5.0.0 8 October 2012

Official release. Update engine to match CatDV 10.0.0. Fix -47 errors when building proxies. Increase memory allocation for helper process and command line. Add REMOTEID, CLIPREF, MEDIASTART, START_SECS, DURATION_SECS fields to exported XML. Use new file chooser. Various other fixes.

5.0b8 1 August 2012

Update engine to match CatDV 10.0b7. Allow \t and \n when setting event markers via the command line. Add support for batch exporting XML and HTML files with all the clips that match a server query in one file, rather than performing a separate export for each task. CatDV XML file now includes metaclip members. Fix related to sending emails. Fix spurious "directory is unstable" warnings. Other minor fixes.

5.0b7 9 May 2012

Release candidate. Fix handling of metaclips returned by a server query. Slight changes to task list details view. Change the location of the worker.xml and workset files so they are in the Application Support or APPDATA directory rather than your home directory. (This simplifies switching between Worker 4 and Worker 5 during testing and avoids cluttering the home directory.) Fix an issue with waiting for directories to settle.

5.0b6 9 April 2012

Start using the new "Slate" look and feel. Improved log viewer with support for displaying the helper log files as well as the main worker log. Prevent helper process under XP due to Windows DLL loading bug that can cause deadlocks. Fixes related to proxy paths. Display error output when running external commands (this is especially useful for reporting progress from the 'curl' command). Fix a timeout that would always occur during long-running commands. Fix erroneous warnings about exceeding client licenes.

5.0b5 31 March 2012

Add proper support for exporting movies from sequences (rather than processing all the source clips within the sequence). Ability to set preconditions on file watcher jobs based on known file types. The group name field now allows variables. Add wizard for editing job schedule. Add checkbox to make it clear that we don't check whether all the contents of a metaclip are online. Add chooser buttons to path map editor and a help button to movie conversion dialog. Miscellaneous fixes related to reading QuickTime metadata, parsing CatDV XML files, exporting movie from a metaclip, exporting XML files, refreshing online status, and fix message about exceeding client licenses. Add development mode (via General tab of configuration) to reduce polling interval and prompt before starting each task.

5.0b4 16 March 2012

Add support for running multiple worker tasks in parallel (with a new drop down on the License tab to control how many worker processes to enable on this machine, up to the number permitted by the license, and the ability to set time outs). Remove the ability to create multiple work sets and incorporate the task list as part of the main window. Add support for scheduling specific times when a job can run. Minor fixes related to metaclips. Include DirectShow (DSJ) exporter on Windows. The command line interface now has the ability to append to a field using "–set FIELD+=VALUE". Fix importing files whose name contains Unicode characters (Windows). Fix exporting files with Unicode names or a long folder name (Mac).

5.0b3 5 March 2012

Add support for running task processor in a separate process (to protect against crashes and provide option for multi-processing in future). Fix "merge MXF files into a metaclip" option but default the option to off as it reduces eficiency of adding clips to an existing catalog. Include Direct Show (DSJ) importer on Windows. Include explicit XDCAM importer and fix handling of XDCAM EX and P2 metaclips. Add a new "allow polling" option to watch folder definitions to process files again if a new file appears with the same name as one processed previously. Improve the "wait for directory to settle" option. Fix proxy path issue on Windows. Allow setting of user columns declared as being checkboxes (use the value "true" to set it).

5.0b2 27 February 2012

Fix it so selected thumbnail size is taken into account. Fix issues with editing presets. Fix doubling up of server query terms. Further tidy up of user interface to make sure it more closely matches order of processing of actions. Ensure "import all files" option works with all importers. Change QuickTime "flattened movie" exporter so it creates fast start movies. Ensure $I variable is set with remote clip id after publishing clips to server.

5.0b1 22 February 2012

Initial beta release built on CatDV 9 codebase. Worker 4 is based on CatDV 8, so this new worker version includes all fixes to the CatDV engine made since CatDV 8.1.12. It includes the following additional major changes: ability to select Xuggle and JMF importers, support for exporting using either QuickTime or Xuggle, a redesigned conversion settings dialog, removal of "pending" state (clips are published to the server immediately rather than being batched up), and more efficient addition of clips to an existing catalog (the worker no longer downloads existing clips in order to append to a catalog).

4.0.6 6 February 2012

Add support for sending SSL email. Fix file not found messages that can occur when using "Move folder" preprocessing action. Ability to specify alternative root for watch folder triggered actions. Added two more extra fields that can be set when publishing clips to the server. Fix broken file paths with multiple colons that can occur on Windows. Fix burnt in timecode on proxy movies. Add support for event fields to the command line interface.

4.0.5 26 November 2011

Fix an issue with building previews under Windows. Monitor file size as well as file and directory modification time to determine whether a file is still being written. Add filter to log file viewer. Please note that PowerPC Macintoshes are no longer supported in this version.

4.0.4 19 October 2011

Remove spurious newline character at end of $x variable. Make @MF writable. Ability to turn off timecode track on exported movies by setting export.timecode=false or preview.timecode=false in Advanced Properties. Ability to delete clip from server for server query triggered actions.

4.0.3 26 September 2011

Fix an issue with multiple @Field=Value responses. Allow a publish interval of 0s to turn off buffering of clips. Fix an issue related to multiple worksets and not publishing changes. Improved diagnostics in the log file. Fix SkeletonMismatchException.

4.0.2 7 September 2011

Fix "no server queries defined"message when using multiple worksets. Fix occasional "duplicate entry on groupID" message when publishing clips to server. Include extra diagnostics to resolve occasional "waiting to acquire lock" errors on Windows.

4.0.1 29 June 2011

Add support for regular expressions, and compatibility with CatDV 9 (saving events and ratings etc.). Add support for copying or moving metaclips based on a server query. Use maximum available resolution when exporting stills from RAW images. Email addresses can now include variables. Set timecode format via timecode.paldefault variable to permit importing EDLs. Task list viewer information panel shows root directory and relative file path separately. Update xalan XSLT library to version 2.7.1. Log viewer has a new button to compress and save the worker log file. Other minor fixes. Added documentation on regular expressions and setting the root for server-triggered tasks.

4.0.0 29 April 2011

General release. Includes ability to override advanced properties if required, plus new date variables $_d, $_m and $_y.

4.0b6 5 April 2011

Second release candidate. Fix an issue relating to watch folders not triggering correctly. Change the 'set timestamp' option so it uses the original file timestamp if there is no record date information. Added new documentation section on "Batch operation".

4.0b5 4 April 2011

First release candidate, now requires valid Worker 4.0 license. Includes improved xml and server query batch support (ability to execute a command on completion of a batch). Fixes relating to exporting stills from NEF files, creating catalogs with variables in the name, and not checking the group when opening a catalog. New $I variable.

4.0b4 28 Jan 2011

Include first version of the new command line interface. Improved support for exporting camera RAW stills. Allow selection of reference movies when building a tape preview movie (by itself this isn't very useful, only if the files are subsequently processed by another tool).

4.0b3 31 Dec 2010

Properly support metaclips with server queries (actions need to apply to the metaclip as a whole, not each individual file within it). Update engine to match 8.1.11. Extend beta expiry.

4.0b2 18 Nov 2010

Improved support for xml batch lists, including path mapping and support for moving or deleting the batch file once all the taks have completed. Improved reporting of timed out old tasks after restarting the worker. Server status display now indicates whether worker is currently connected to the server or not. Update engine to match CatDV 8.1.9. Other minor fixes.

4.0b1 26 Oct 2010

Fetch user field names from a field definition set on the server and include these in exported XML files. Up to four extra fields can now be set when publishing catalogs. Ability to test watch actions directly from the worker config window, reporting how many candidate clips might be found in a watch folder or by a server query. New enable checkbox to simplify enabling or disabling watch actions. Support for unlimited server query terms. Support for mapping media file paths when performing a server query (eg. to handle difference between Mac and Windows file paths). New option to perform two-pass conversion when exporting QuickTime movies, allowing two sound tracks with different settings (for example, one 5.1 and one stereo mix down) to be created. Old legacy conversion presets are no longer shown by default (click on the Format: label to show these if required). Ability to send success and failure notification emails to different recipients.

3.1.1 29 Aug 2010

Update engine to match CatDV 8.1.5/8.1.6, including improved extraction of frame rate and audio bit rate, improved xml batch file support, and extraction of metadata from AVCHD .mts files. Improved memory usage when processing RAW files and performing server queries. Better reporting of configuration errors. Fix exporting of thumbnails from server queries.

3.1.0 9 Jun 2010

Updated engine to match CatDV 8.1.3, including support for Broadcast WAV timecode and other improvements when extracting metadata. New $U and $R variables. New transcoding options to set the modification time of exported files to match the record date, to rebuild and overwrite a file if the destination exists (normally existing files are skipped), and to set the position of burnt-in text. New scene detection option to create markers rather than subclips. Fix an issue with filenames containing Cyrillic characters. Fix setting of group after disconnect and reconnect to server. Don't import hidden folders.

3.0.8 20 Mar 2010

New option to truncate generated filenames so they don't exceed a given length. New option to copy the input file during the preprocessing stage. New variables $1 to $9 to pull out parts of a file name. Ensure $p, $q and $Q variables are available earlier. Fix an issue with duplicating server watch items. Automatically reconnect if the server is restarted to avoid "stale session" errors. Update fields like Media Path earlier so they are included when exporting XML. Expanded documentation section on "Pre-processing".

3.0.7 27 Feb 2010

New option to turn off automatic linking of MXF files based on UMID if required. Ensure that tasks don't continue if the preprocessing step failed. Support for processing a batch of clips given by a CatDV XML file. Fix an error about &#0 not being valid XML.

3.0.6 12 Feb 2010

Check whether a file is locked before attempting to process it. Allow group in which new catalogs will be created to be specified on a per-task basis.

3.0.5 2 Feb 2010

Bring engine up to synch with CatDV 8.1b. Add TIFF/RAW importer. Add support for exporting stills from event markers. Add support for exporting P2 metadata. Burnt in text now supports variable expansion. New $Q and $P variables. Added POSTER and REMOTEID fields when exporting clips to XML. New documentation sections on "How file naming works" and "Exporting stills". Perform file pre-processing stage *before* opening the catalog so variables such as $p, $q, $f, $g, $P and $Q are now available for use in catalog names.

3.0.4 11 Dec 2009

Fix to ensure compatibility with Java for Mac OS X 10.6 Update 1. Synchronisation fixes related to multiple workers working on the same task list. Fix 'move' preprocessing action. Preserve selection in task list when tasks execute. Ensure aspect ratio is set when publishing to server. Fix occasional StringIndexOutOfBoundsException. New $G and $M variables.

3.0.3 27 Oct 2009

Support for exporting clips as an XML file, possibly translated with an optional XSLT file. Support for executing an external command, passing it parameters via XML and processing the return value to update clip fields. Optional support for polling the server and repeatedly processing the same clip if required. Other cosmetic improvements to help text, server queries, and sending email.

3.0.2 21 Sept 2009

Improved support for parametrising commands: allow variables within file paths, email messages and when naming catalogs; additional predefined variables; able to include clip values within expressions; allow special characters like newline or tab within an expression. New preprocessing option to copy or move P2 folder structure. New option whether to import all types of file (including unsupported files) when importing media files. Better control over thumbnail creation. Ability to copy or move a clip to a new catalog when performing server tasks. Use new APIs when reading QuickTime metadata. Fix issue with creating previews under Windows. Changed the way configuration errors are reported. Minor fixes to status display.

3.0.1 17 Jul 2009

Improved P2 and XDCAM metadata support. Read enclosing folder name when concatenating folder and file name. Fix “out of range value”that sometimes occurred when publishing clips. Fixes relating to non-drop frame NTSC timecode. Add a #Clips column to the task summary, showing how many clips were processed (this will normally be 1 except for scene detection or related files in a P2 folder).

3.0.0 27 Jun 2009

General release. New option to export still images. Display a summary of tasks in the task viewer window. Compatible with Java for Mac OS X 10.5 Update 4.

2.0.3 23 Feb 2009

Updated ingest engine to match CatDV 7.0.6. Allow creation of tape- or path-based previews (as well as standalone exported movies). Protect against creating folders in /Volumes. Other minor fixes.

2.0.0 13 July 2008

General release. Added openLocalCatalogs flag. Updated engine to CatDV 7.0 final.

1.0.0 24 November 2006

First release candidate. Implement sending email notifications. Minor user interface fixes. Documentation updates.

Overview

The CatDV Worker Node is an automated version of the media processing engine found in the CatDV Pro application. It has two main components, the file watcher that watches one or more watch folders looking for new files to process, and the worker, a background thread which waits for and then processes the next available item from a queue of tasks.

The worker node must be configured to define which watch folders it monitors, what operation(s) to perform on these files, and how to access the CatDV server.

To configure the worker you specify:

- common definitions, such as the host name of the CatDV Workgroup or Enterprise Server, or details of an SMTP server for sending email notifications
- the location of the task queue file that the file watchers will add items to and the worker thread will remove items from as it processes them
- a definition of how items are added to the queue, ie. which folder to watch and any filter conditions that the file must fulfil to be processed (such as its filename matching a particular pattern, or it containing media in a particular format)
- a definition of what operations to apply to matching files.

Supported operations that can be performed include the following:

- analyse the file to extract technical metadata such as the video and audio format, sample rate, timecode etc.
- perform automatic scene detection to create separate clips for each scene

- create a CatDV preview movie from the media, or convert it to another format using an arbitrary QuickTime exporter and codec
- publish the clips (ie. details of the file and any scenes that were detected) to the CatDV server
- generate or update an HTML index page listing the processed clips
- send an email notification to selected users notifying them of the new file(s)
- execute an arbitrary operating system command, such as uploading the files to an FTP server, running a shell script or batch file, or anything else.
- move or delete the original media file once it has been processed or converted

The combination of a watch folder, filter condition and job definition is known as a “watch action”. You can define as many watch actions as you want, all with different filter conditions, to automatically process different kinds of file in different ways.

The combination of a task queue file plus a set of watch actions that will cause items to be added to the queue is known as a “work set”. The work set represents a queue of tasks to be performed (or that have already been performed recently).

Quick Start Guide

This section provides a quick tutorial, covering the basic steps you need to go through to start using the worker node:

1. Launch the CatDV Worker Node application. It should automatically bring up the Edit Configuration window, otherwise press the Edit Config button.
2. Go to the “License” tab and enter your registration code. (To do this, select and copy the entire contents of the email message you were sent then simply press the Paste button.)
3. If you use the CatDV Workgroup or Enterprise server, enter the server hostname and other details under the “CatDV Server” tab. (If you use the Enterprise edition of the server you should create a special user for the worker node to use and enter the username here.)
4. If you want to send email notifications you need to enter details of the SMTP mail server you wish to use under the “Email Setup” tab.
5. Go to the “Watch Actions” tab and click the “+” button to define a new watch action.
6. First, define what files the action applies to. It can be triggered by a file occurring inside a particular watch folder or by dropping a file into the Worker Node application window, and it may be restricted so it only applies to files that meet certain filter conditions. For example, you could enter “*.mov” as a filename pattern, or “44.1” in the audio format field. (Alternatively, you can define a server query and clips that have been saved to the server will be processed instead, for example whose status is “Publish via FTP”.)

   If you don’t want to check a particular condition just leave the fields you’re not interested in blank to indicate “don’t care”.
7. Then, define what actions you want to apply to the matching files. Enter a job name to describe the action (eg. “Convert audio to 48 kHz”) and complete other fields under Job Details and Post-Processing as required. Many of the operations correspond to commands in the CatDV application. Analysing media corresponds to importing a media file and generating clips, movie

conversions correspond to the Export As Movie and Build Preview Movies commands, and so on.

8. Press the "+" button on the Job Details tab to define as many movie conversions as you want. A single input file might be processed to create multiple versions at different resolutions for example. For the format, choose New Conversion Preset, give it a descriptive name then select which exporter to use and press Customise to adjust its settings. You can also choose a preset you have previously defined.
9. Repeat steps 6 through 9 as required to define as many different watch actions as required, or use the "Duplicate" button to use an existing watch action as a template and then edit the copy.
10. When you are finished, press Apply to re-initialise the worker node with the new settings and start processing files.

If you specified one or more watch folders they will be scanned initially after a short delay, and then repeatedly after the period that you configured. Alternatively, if you enabled file dropping, then you can drag and drop files from the Mac OS X Finder or Windows File Explorer into the main Task Queue area to submit them for processing.

User Interface

When you launch the worker node application it displays its main window, which summarises the current status of the worker and has commands to let you edit the configuration, display the detailed status of a particular task, and view the log file.

On the Mac the worker node continues to run in the background even when the main window is closed. Under Windows it terminates when you close the window.

Three distinct processes run concurrently and independently in the worker node, the File Watcher (which periodically scans the watch folder(s) looking for new files), the Server Watcher (which performs queries against the server looking for clips that are flagged for processing), and the Worker itself (which fetches tasks from the task list and performs them). The main window has check boxes which you can use to enable or disable these processes and also displays what they are currently doing.

Task Queue

At the heart of the worker node operation is the task list, a combined queue of tasks waiting to be performed and list of tasks that have already completed. Each task represents a clip or file that needs to be processed and a job to be applied to it.

The main window shows your current tasks in a table, with one task per row, whether completed or queued waiting to run. Each task has details of the input file and the job to be applied to it, together with information such as when the status was last updated, when the file was last seen (for example, if it the file is on a removable or network volume and is currently offline), or whether the task entry was created by the current running instance of the worker node or not (either because the software has been restarted or because another instance on another machine is using the same work file).

A task can be in one of several different states:

Queued — When a new file appears in one of the watch folders a new task is created and queued. When the worker runs it will start executing the first available task whose status is 'queued'.

Running This means the task is currently being executed. Note that several worker nodes on different machines could be working on the same task list, so it's not necessarily being executed by the instance on this machine. (In practice though, if the instance is not the current one the most likely explanation is that an earlier instance was terminated prematurely and the worker node software has been restarted.)

Failed The task has completed but an error occurred. See the Information field for further details. If you correct the problem (eg. by starting the workgroup server, if it was previously down) you can resubmit the task so it is performed again.

Complete The task has successfully completed

Hold If you want to temporarily defer tasks (for example, to allow tasks that were queued later to be executed first) you can mark them as being on hold.

Offline This status is similar to tasks being on Hold. Tasks that were previously queued are automatically set to Offline when the worker node starts up if the file they relate to is no longer available (for example, because a server volume is temporarily offline).

Skipped Some jobs (such as importing an image sequence or a P2 metaclip) process multiple files in one go. If a file was queued for processing but has already been processed by another job it is marked as Skipped.

You can resubmit failed tasks (for example, after restarting the CatDV server), hold or resume tasks, or delete tasks using the relevant buttons at the bottom of the task list.

Tasks have a priority which affect the order in which they are done. The job definition defines a default priority but you can adjust the priority of specific tasks in the queue if you want to bring one forward.

Tasks are normally hidden after 3 days (unless you check the Show All box) and deleted altogether after 14 days.

Configuring the Worker Node

When you first launch the worker node software a message will be displayed because you don't have a valid configuration. Press the Edit Config button to edit the settings. You will be taken through various screens and tabs for the different parts of the configuration:

General settings

Under the **General** tab you can choose your preferred format for displaying dates and times. You also specify how catalogs which are auto published to the server are named.

You need to enter a valid worker node license by entering the user name and activation code under the **License** tab. The easiest way to do this is select the email message you were sent (contact sales@squarebox.com if you don't have a copy), copy the entire text, then press the Paste License button.

Under the **CatDV Server** tab are settings that relate to the CatDV Workgroup or Enterprise Server. To automatically publish clips to the server you need to enter the server name (and optional port number) in the Server URL field.

If you are using the CatDV Enterprise Server enter the user name, password and group name (available from the CatDV Server Admin panel) that you want to own any catalogs that are automatically published by the worker node.

So that the correct names for user-defined fields are shown when defining server queries or settings clip fields, if you are using the Enterprise server you can enter the name of the field definition set you are using and press the Connect (or Re-Connect) button to load those field names from the server.

Publish Interval defines how long to wait before a clip is published to the server. Using this setting allows several files to be batched up and published in one go, which is more efficient but introduces some latency before other machines on the network will see the files.

To send email notifications whenever a file has been processed you need to have access to an SMTP server that will accept messages sent from the worker node. Under the **Email Setup** tab enter the server hostname, the From address that should appear in the message and a default subject (you can specify different subjects in different watch actions). You can also enter a user name and password if the server requires authentication.

In the same way as publishing clips to the server can be batched, email notifications are also batched so no more than one message is sent in a given period. You specify the recipients for normal notifications as part of each specific watch action definition, but if an error occurs (for example, an error connecting to the CatDV server) you can enter the email address of the administrator user(s) to be notifed.

The configuration needs at least one work set definition to be valid. You define work sets and watch actions under the **Watch Actions** tab.

Creating work sets and watch actions

In the current version of the worker you can only have a single work set. The work set includes the name of a task list file. This file is important as it contains the queue of tasks, both completed and currently running, and other information. In most cases you can use the default task list file but if necessary you can specify a different task location by clicking on the name to display a file chooser. The file name you choose should have extension .xml.

If you will run the worker node under your user account a good place for the task list file is in your own home directory, as you will need read and write access to both the file and the directory that contains it.

You can define any number of watch actions in your work set, each of which defines a watch folder and tasks to be performed on any matching file that appears in the watch folder. Press the '+' button to add a new watch action or use 'Duplicate' to copy an existing watch action (you can then edit the copy without having to enter common settings again).

Editing watch actions

A watch action has two distinct parts. A definition of which clips or files should be processed (based on conditions such as where they are or what type they are) and a definition of what operation(s) should be applied to those files.

An action can be triggered based either on files appearing in a watch folder (or being dragged onto the worker icon), or based on clips saved to the CatDV server that match particular query conditions. If an action is triggered by the appearance of a file it must first be imported using one of CatDV's importers (for example, analysing the file using QuickTime to extract metadata such as the file format, duration, timecode, etc.) whereas for a server-triggered action the clips have already been imported (either from a CatDV client or perhaps by another worker task) and so the file pre-processing and import stage is omitted.

The ‘General’ tab defines the name of the job, its relative priority, whether to send an email notification on completion of the job, and whether the job is triggered by a file or by a server query.

For file-triggered tasks, the ‘Trigger’ and ‘Conditions’ tab define the files to be processed, while the ‘Pre-processing’, ‘Conversions’ and ‘Post-Processing’ tabs define the actions that will be performed. For server-triggered tasks a ‘Server Query’ tab defines the query used to select clips for processing, while the ‘Conversions’ and ‘Post-Processing’ tabs are as before.

File-triggered tasks

The ‘Trigger’ tab defines whether files are processed because they appear in a watch folder or when they are drag and dropped. The ‘Conditions’ tab defines various filter conditions that the file must match for this watch action to be performed. Finally, the ‘Job Details’ and ‘Post-Processing’ To use watch folders the root watch folder must be specified. You can choose whether to only analyse files that appear in the watch folder or recursively in any subdirectory.

Alternatively, you can also enable drag and dropping of files directly onto the worker node application window. You must enable either the watch folder or dropping files (or both) or the watch action won’t be active (though there may be situations where it’s useful to disable a watch folder so it isn’t scanned without having to delete the entire watch action definition).

To avoid processing a file while it is still being captured or copied a settle period can be defined. Files are ignored if they are less than a given number of seconds old. The check interval defines how often the watch folder is scanned. Increase this period if you are recursively large directories on a slow external volume. (These fields can be left blank to use their default values).

Watch conditions

If you want to limit the processing to files of a particular type you can do so by setting various filter conditions.

The simplest and most efficient test is to only include files whose name matches a particular pattern, or which does not match the ‘exclude’ pattern. The filters can either be simple patterns, such as ‘*.mov’ or a complex regular expression such as ‘/[A-Za-z]+\.mp4$’ if the regex box is checked. Case is normally ignored when matching these patterns.

Other tests involve analysing the file by trying to open it as a media file (using QuickTime). You can choose one of various basic types (eg. QuickTime movie, AVI movie, MPEG movie, Still image, Non media file) from the drop down, or for media files add additional criteria based on the movie duration in seconds, the audio and video format, or testing for arbitrary track types.

In each case, you can check for the presence or absence of a particular property, and if you leave a field blank the condition will not be checked.

For video and audio tracks a description of the format (like the Video and Audio fields of CatDV) is checked for the presence of the text you type in. For example, Video might say “Photo – JPEG (480x320 30.0fps)” while the Audio description might be “MPEG Layer-3 Audio (44.1 kHz, stereo, 16 bit). Any substring will match, so you could type “JPEG” or “30.0fps” or “Photo – JPEG (480x320” in a condition.

To test for the presence of arbitrary track types, enter the type and subtype codes as displayed in the QT Tracks field of CatDV.

Pre-processing

You can pre-process files by moving or renaming them (including changing the file extension or Mac file type/creator) prior to importing or converting the files, for example to give files a unique name early on in the process if they would otherwise be known as something generic like C0001.MXF or IMG0001.JPG. It can also be useful when dealing with long filenames or those containing accented characters, which can cause problems in QuickTime under Windows.

There are several renaming options available. The file can be named by concatenating the folder and file name, by using a timestamp based on the current date and time, or by moving files to a destination folder and numbering the files consecutively so they get a unique name that doesn't clash with files already there.

The naming drop down is used to specify both the destination folder (when moving files) and the file name. Some combinations are not permitted, for example "Rename file" and "Use enclosing folder", as the latter is used when moving a file to another folder but keeping the original filename, and the former is used when renaming a file but leaving it in its original folder. See "How file naming works" below for further details.

The destination field is used both to specify the destination folder (when copying or moving a file) and to enter the filename pattern when expanding variables. These can be combined, for example you could specify a destination of "/MediaArchive/$Q_$1_2e" and then pass in a file called "/Volumes/XYZ/A001/A001_B02_1234_HI.MOV" to give "/MediaArchive /XYZ_A001_B02.MOV"

Most pre-processing operations refer to a single file but when dealing with things like P2 volumes you can copy or move the entire enclosing folder structure when the action is triggered. You need to specify a root folder (see below) and it's the top level folder within the root that will be moved. For example, if the action is triggered by a file "/Folder/Subfolder/File1.mov" within "/Root" then /Root/Folder (and all its contents) will be copied or moved.

Importing media files

After pre-processing, the file needs to be "imported" to extract technical metadata (including its duration, format, timecode, and the tape name, if present in a timecode track) and create a clip record before you can transcode it to another format or publish it to the server. Optionally, scene detection is performed, using either DV-based start/stop recording information or visiual scene analysis, to create separate subclips. A separate thumbnail image for each clip is created of the specified size.

The following importers are available:

AVI/WAV File This uses CatDV's built in parser for AVI and WAV files

ASF/WMV File This uses CatDV's built-in parser for Windows Media files.

CatDV Catalog Open a .cdv catalog file, with the option to replace an existing catalog on the server in its entirety (may be used to synch catalogs from one machine to another)

CatDV XML File Import a CatDV XML file in "immediate" mode, see below

CatDV XML File (Quick) As above, but do a quick import without analysing the media files

Generic File This imports a file as a non-media file with a generic icon and no analysis of the media format (such as duration)

Image Sequence If a folder contains consecutively numbered still images the worker can import these as a single image sequence clip.

JQT MOV/MP4	This uses CatDV’s built in parser for MOV and MP4 files
MPEG Media	This uses CatDV’s built in MPEG parser to determine technical metadata for MPEG-1 and MPEG-2 program and transport streams
MXF File	This uses CatDV’s built in MXF parser
P2 Metaclip	This uses CatDV’s built-in MXF parser and processes all the related files for a P2 clip in one go (video, audio and clip xml details are all stored in separate files but the P2 importer creates a single ‘metaclip’ that represents the clip as a whole)
PDF Document	This uses CatDV’s built-in PDF parser
R3D File	This uses CatDV’s built-in parser to read metadata from a single RED .r3d file
RED Metaclip	Import a RED clip, creating a single metaclip that includes multiple spanned .r3d files and associated files such as reference movies in the same folder
QuickTime Media	This is the default importer for media files in any format supported by QuickTime, including movies, still images, audio files and more.
TIFF/RAW Image	Import camera raw files (eg. CR2, DNG, NEF, MOS etc.) and TIFF images
XDCAM Metaclip	Import XDCAM, XDCAM EX and XDCAM HD folder structures and create a single ‘metaclip’ that wraps up all the files associated with a clip

In most cases you can use the “Automatic” setting to automatically determine the correct importer to use for common media file types.

If you choose the “Import all files” option then other types of file that the worker doesn’t know about are imported as generic files. Additional, if a known file type is corrupt and importing it normally gives an error, the worker will fall back to importing it as a generic file.

If you choose the “Interpret batch files” option then the contents of known batch and log files (with extension .cmx, .edl, .ale, .txt, .tsv, .xml, .srt, .tab, .dmp, and .fcpxml) are parsed using the corresponding CatDV importer (eg. the CatDV Pro “Import As Tab-Separated Text” command). This can result in many clips being produced (one per entry in the batch file), including offline clips.

If you choose the create MXF Metaclips “Based on folder structure” or “Based on MXF UMID” option then Avid, P2, XDCAM and similar formats where one clip can consist of separate files will result in one “metaclip” rather than separate individual clips in your catalog.

To create a clip per file you should therefore check the “Import all files” option, uncheck “Interpret batch files” and choose the “Don’t create metaclips” option.

In most cases you should choose “Based on folder structure” rather than “Based on MXF UMID” as this is more efficient, and gives more consistent results when running with multiple worker processes. You should only choose the MXF UMID option if you are working with Avid MXF files or need to merge spanned P2 clips into a single metaclip.

Automatically keeping catalogs in sync with a disk or folder

A common use of the worker is to automatically catalog an entire volume or folder and keep the catalog up to date as new files are added. To do this efficiently and provide more flexibility and control over the process this is normally done by breaking it down into a number of related tasks:

i) Use Quick Folder Scan to quickly scan the volume and create generic stub clips for each file, permitting files to be accessed or archived via CatDV as soon as they appear on disk. Set a clip status field to ‘Awaiting analysis’.

ii) Perform a server query to search for files of status ‘Awaiting analysis’ and perform media analysis on these to extract thumbnails and technical metadata. This will add each file to the task queue for separate processing but can run as a low priority background task. On completion, update the status to ‘Analysed’.

iii) Use another server query task to transcode the files and create proxy versions as required, either automatically (perhaps all video clips) or manually (based on the user selecting the clips they want proxies for within CatDV and ticking a checkbox to say ‘Create proxy’). On completion set the status to say ‘Proxy created’.

iv) Finally, you can have a bulk server query that tests whether all the files listed in the catalog are still available online and/or whether proxy files exist, and can update other clip fields to say ‘Original file exists’ and ‘Proxy file exists’. This provides the converse to Quick Folder Scan and will identify clips that exist in the catalog but have since been deleted from disk.

v) If desired, a separate task could be used to automatically delete both the clip from the catalog and the corresponing proxy file if an original file on the monitored folder is deleted. This isn’t recommended however. To avoid the disk of losing metadata deleting clips is usually better done manually (files may be offline only temporarily, if a folder is accidentally moved or a drive isn’t mounted for example, and you wouldn’t want to delete your clips in those circumstances).

These steps can be combined if required, for example combining ii) and iii) to create proxies at the same time as the files are analysed.

Unless the volume or folder being monitored is very small you may want to place clips into different catalogs based on the folder hierarchy. For example, if you have a watch folder on /Volumes/SAN and files are organised into a hiearchy based on customer name and project number then you might want to create a separate catalog for each project. The $P variable contains the path of each file relative to the watch folder root but as the project may contain subfolders we want to use the first two components of $P as the catalog name. We can do this using regular expressions, first to convert Windows paths to Unix-style paths if necessary then pull out the first two groups of characters between a forward slash. /[^/]+/ matches a / followed by one or more non-/ characters followed by another /. The Quick Folder Scan action might therefore publish to catalog name:

```
SAN$P{s,\\,/,g}{s,(^/[^/]+/[^/]+)/.*,$1}
```

When checking whether clips are still online at step iv) we can search for clips where the catalog name starts with “SAN”.

(You might also want further statuses such as ‘Ready for logging’ and ‘Logged’ and move clips or files to another catalog or library location once they are logged, but this depends on your desired workflow.)

CatDV XML batch files (v1)

You can submit a batch of media files to the CatDV Worker for processing by providing a list of file names (optionally with metadata) in the CatDV XML file format. You can create an example XML file in the format the worker expects by exporting clips from CatDV using File > Export As > CatDV XML Batch File, though not all the fields in that file are used when you reimport a CatDV XML file.

The main field you need to specify is the MEDIAPATH of the file to import but you can also specify additional metadata, for example:

```
<CLIPS>
      <CLIP>
           <MEDIAPATH>/Volumes/Media/Folder/File1.mov</MEDIAPATH>
      </CLIP>
      <CLIP>
           <NAME>My clip name</NAME>
           <MEDIAPATH>/Volumes/Media/File2.mp4</MEDIAPATH>
           <NOTES>This is a clip with metadata</NOTES>
           <USER1>User field one</USER1>
           <TAPE>Tape001</TAPE>
           <BIN>Bin name</TAPE>
      </CLIP>
</CLIPS>
```

There are two ways to import a CatDV XML file, in either “batch” or “immediate” mode. For further details, see the sections on Batch Operation and Offline Workflows below.

If you are submitting a batch of files to the worker for processing, possibly involving a transcode of each file, then you need to create a CatDV XML batch file (version 1) and import it in “batch” mode to create a separate worker task for each file. If you are importing a set of clips in one go (or updating existing clips) then you should use the new CatDV XML file format version 2.

New CatDV XML files (v2)

While version 1 XML batch files will continue to be supported, the new CatDV version 2 XML file format is more modern and more flexible and properly supports sequences, subclips, markers, metaclips, thumbnails, and more.

To import a clip by analysing the media file but specifying certain metadata fields you can write:

```
<catdv version=’2.0’>
      <options analyseMedia=’true’ updateOrAdd=’add’ />
      <clip>
            <name>My clip name</name>
            <notes>This is a clip with metadata</notes>
            <tape>Tape001</tape>
            <bin>Bin name</bin>
            <metadata>
                  <field index=’1’ name=’User 1’>User field 1</field>
                  <field index=’23’ type=’multi-picklist’ name=’Weather’>
                        <value>Rain</value>
                        <value>Cloud</value>
                  </field>
            </metadata>
            <media>
                  <filePath>/Volumes/Media/File2.mp4</filePath>
                  <metadata>
                        <field key=’Author’>Joe Smith</field>
                  </metadata>
            </media>
      </clip>
</catdv>
```

You can use XML to delete an existing clip on the server:

```
<catdv>
      <options updateOrAdd=’delete’ searchBy=’remoteID’/>
      <clip>
            <remoteID>623462</remoteID>
      </clip>
</catdv>
```

The following example demonstrates creating a clip and a subclip referring to the same source media object, without performing media file analysis:

```
<catdv>
    <clip>
        <name>File.mov</name>
        <type>Movie</type>
        <notes>Master clip</notes>
        <in seconds="0.0">0:00:00:00</in>
        <out seconds="10.0">0:00:10:00</out>
        <timecodeFormat>25.0</timecodeFormat>
        <media id='0'>
            <filePath>/Volumes/Media/Offline/File.mov</filePath>
            <video>Apple ProRes 422 (HQ) (1920x1080 25.0fps)</video>
            <audio>24-bit Integer (48.0 kHz, stereo, 16 bit)</audio>
        </media>
    </clip>
    <clip>
        <name>Subclip</name>
        <type>Movie</type>
        <in seconds="5.0">0:00:05:00</in>
        <out seconds="7.0">0:00:07:00</out>
        <media id='0'/>
    </clip>
</catdv>
```

Notes on the new XML file format:

- Although none of the fields are mandatory when importing a clip, in order to create a useful clip (without relying on CatDV or the Worker analysing a media file) as a minimum you should ideally provide all the following: <name>, <type>, <timecodeFormat>, <in>, <out> (or <duration>), plus a <media> object with <filePath>, <video> and <audio> elements.
- The following timecode formats are supported: 1.0, 10.0, 15.0, 23.98, 24.0, 25.0, 29.97, 29.97 nd, 30.0, 50.0, 59.94, 59.94 nd, 60.0, 100.0.
- <clip> and <media> (and <importSource>) elements can include an id attribute, which allows multiple subclips that refer to the same source media object to be represented. It also allows sequences to refer to their source clips. (For <media> and <importSource> the id is an integer, while <clip> can use any unique string.)
- There is a consistent syntax for user-defined and media metadata fields, paving the way for general metadata fields in CatDV Server 7
- Multi-grouping fields take multiple values by providing multiple <value> elements rather than using a `&#10;` new line delimiter.
- Markers have separate elements for <name>, <category> and <description>, which allows for additional custom marker columns to be added when using CatDV Server 7.
- Time code values and timestamps (and also file sizes, data rates, and clip types) are stored both as an internal representation and a user-friendly formatted representation. For example, `<in seconds=’1.5’>0:00:01;15</in>`, `<modified timestamp="1420560808038">Jan 6, 2015 16:13:28</modified>`, `<fileSize bytes="1116192000">1.04 GB</fileSize>`, etc.

On export both representations are used but during import the internal representation is preferred.

- Thumbnails can be embedded in the XML file, stored as base64 encoded JPEG data (which must not exceed 64KB per thumbnail). This allows a complete clip representation including thumbnails to be imported into CatDV via XML, even if the media file is offline.

- For examples of other features such as as <sequence>, <metaclip>, and <importSource> elements please export a version 2 XML file from CatDV or the Worker. A DTD can be provided on request.

When importing an XML file you can provide an <options> element to control how the XML file is interpreted on import and whether to update existing clips or add new ones:

- You can control whether to analyse the media file on import (to determine the clip duration, technical metadata, extract metadata from the file name or a sidecar file) using `<options analyseMedia='true'/>`. By default, media files are not analysed and all the technical and other metadata is specified in the XML file.

- You can specify whether to update existing clips stored on the server or insert new clips using an element such as `<options updateOrAdd='updateOrAdd' searchBy='filePath'/>`.

- Valid update modes are 'add' (this is the default and results in new clips being added even if they already exist), 'addIfNew' (add a clip if it doesn't already exist, otherwise do nothing), 'updateOrAdd' (update an existing clip if it exists or add a new one), 'updateOrIgnore' (update an existing clip if it exists, otherwise do nothing) and 'updateOrFail' (update an existing clip if it is found, otherwise report an error).

- The special update mode 'delete' can be used to delete existing clips on the server.

- Existing clips can be searched for by 'remoteID' (the unique integer database id of a clip), 'clipRef' (the user-settable clip ID), 'filePath', 'clipRefAndPath', 'catalogAndPath', or 'catalogAndClipRef'. Note that only remoteID is guaranteed to be absolutely unique so any of the other options could result in multiple clips being updated (or deleted).

- Normally updating or adding clips always applies to the server, both when the XML file is processed by the Worker or by CatDV/Pegasus desktop clients. Use `<options updateLocal='true'/>` to update clips in the currently open catalog in memory in the desktop client.

- If you update an existing clip via XML only those fields which are specified in the XML file are updated and other fields keep their original values. You can update user defined and media metadata fields, standard clip fields such as name, notes, status, tape, bin, record date, and rating, markers, plus the media path and archive status, via XML. With care you can also update the clip in and out points, type, poster, and clipRef. When updating markers, the markers you specify are normally *added* to any existing markers unless you specify `<markers replace='true'>…</markers>`

- When updating clips you can choose whether to add new thumbnails or delete existing ones first and replace them by specifying `<thumbnails replace='true'>`.

- When updating an existing clip you can specify how to combine the new value for certain elements (`<notes>`, `<bigNotes>`, and `<field>`) with the existing value by adding a merge attribute with values: 'replace' (overwrite the old value), 'prepend' (prepend the new value to an existing value), ‘append’ (append the new value), ‘default’ or ‘true’ (don’t overwrite an existing value but set it with new value if it is currently not set), and ‘merge’ or ‘combine’ (split the values into lines and merge the two sets of lines). For example `<notes merge=’true’>Under review</notes>`

- You can have several <options> elements, each of which will apply to all subsequent clips in that file until the next <options> element is encountered.
- When importing clips via XML the import source is normally set to describe the XML file the clips were imported from. It is possible to override this by providing an <importSource> element and specifying `<options importSource=’true’/>`.

Metadata extraction rules

Many workflows involve importing files that have been named according to strict naming conventions and where the file name or file path is structured to represent elements such as the series and episode of a show, the type of shot, the camera used, the scene and take number, and so on. While it is possible to use the Publish tab to set clip fields based on worker variables and regular expressions applied to the file path can be awkward when the file naming conventions become more complex.

A new and more powerful metadata extraction mechanism has been added which involves providing an external mapping file that lists patterns to look for and then which metadata fields to set when importing a file if those patterns are found in the file path.

You can specify a file containing metadata extraction rules on the Options tab of the worker config. The format of the file is lines of the form: *pattern* **tab** *field id* **tab** *value to set* **newline**. The pattern can either be a simple text pattern or a regular expression. Blank lines and comment lines starting with a semicolon are ignored.

If the pattern contains nothing but upper or lower case letters and numbers then it’s taken as a token that is looked for in the file name. Tokens in the file name must match case and be delimited by a space or punctuation character. For example, the file name “FB_S2_XYZ_Scene01.mov” would match FB and S2 but not s2 or XY or 01.

You might have the following metadata extraction rules and user field 1 would be set to “Football” and user field 2 to “Season 2” when the file above is imported.

```
FB    U1    Football
BB    U1    Basketball
S1    U2    Season 1
S2    U2    Season 2
```

If the pattern contains anything other than basic alphanumeric characters it is taken to be a regular expression, where certain symbols such as . + * ( ) [ ] ^ $ \ { } etc. have special meanings.

If the pattern starts with lower case ‘m’ or ‘n’ and followed by a punctuation character such as comma, semicolon or slash then it is taken to be a regular expression that applies to the entire file path or just the file name respectively, and the specified character is the delimiter at the end of the regular expression (as you might want to use something other than / when matching file paths). In either case, you can follow the regular expression with ‘i’ to ignore case. The value of the first subexpression within parentheses is available in $1, the second set of parentheses as $2 and so on, and $0 refers to the full pattern.

For example, you might have

```
n/Scene(\d+)/    U3    Scene number $1
```

This would look for a sequence of digits occurring after the text "Scene" anywhere in the file name, and if found store the scene number in user field 3.

You can use a more complex pattern like `m,/([^/]+)/[^/]+$,` to extract the parent folder into $1, using comma as the regex delimiter and where forward slash is the file path separator on Mac OS X and Unix. [^/]+ means any sequence of one or more characters which aren't a slash and '$' anchors the search to the end of the file path.

If you omit the pattern then you can have several substitutions triggered by the same regular expression, saving on evaluating the expression more than once. For example

```
m,^/Volumes/Media/([^/]+)/([^/]+)/,    U2    $1
                                       U9    $2
```

would store the first folder after /Volumes/Media in user 2 and the second folder in user 9.

You could use a similar mechanism to pre-populate a set of metadata fields such as customer name, contact details etc. based on which customer folder the file is in.

Detailing all the capabilities of regular expressions is beyond the scope of this document but if you require help in defining metadata extraction rules (or developing worker scripts more generally) then your systems integrator or our professional services team will be able to help.

To simplify working with a mix of Mac and Windows paths all file path separators are converted from backslashes to Unix-style forward slashes so regular expressions should only check for /, even on Windows. Also, you can define root names and match for these as well, so the pattern above could be written as

```
#root MEDIA /Volumes/Media
#root MEDIA \\server\media
m,^MEDIA/([^/]+)/([^/]+)/, U2    $1
```

You can organise rules into separate files and import another file using

```
#include "/path/to/file.txt"
```

Exporting watch definitions

It is possible to export watch action definitions (and also conversion presets) by pressing the Export button. This will create an XML file with extension .catdv that can be dragged into another worker (either into the main window or into the watch action list in the configuration dialog) to install the watch action on that machine. If the watch action includes movie conversion steps the required conversion preset definitions are automatically included in the .catdv file.

When you install a watch definition you are prompted whether to replace an existing watch action with the same name or keep both and rename the old one.

You can parameterise .catdv settings files so that when a user installs them they are prompted to specify the values for certain parameters, such as the watch folder location, an FTP password, or YouTube account details, as these will differ from installation to installation.

To parameterise the file you should include text in the format {{Parameter name{Type information}:Default value}}. Everything between {{…}} will be replaced with the appropriate value entered by the user. For example,

```
 <convert previewDir="{{CatDV proxy folder{DIR}:/Volumes/CatDVProxies}}" … >
```

or

```
 <job … md5Checksums="{{Calculate MD5 checksum?{Yes=true,No=false}}}" … >
```

You can omit the type information and default value. Omit the type information for plain text fields, use {FILE} or {DIR} for a file or directory chooser, or use a comma-separated list {Option1,Option2,…} for a drop down pick list of values. If required these options can have a different display value from what is inserted into the final definition, for example with {Red=0xf00,Green=0x0f0,Blue=0x00f,Yellow=0xff0} the user could pick Green from the drop down list when installing the action but 0x0f0 would be inserted into the definition.

Server-triggered tasks

For server-triggered tasks you use the 'Server Query' tab to specify a query that will be performed to find matching clips for processing. You can also specify a check interval, and whether only to process clips which have a file which is currently online and accessible. If you specify a root folder, only files within that folder are processed, and that folder is used when calculating relative paths (see below).

Queries are defined in the same way as advanced queries in the CatDV client. If you specify multiple query terms (for example, based on clip status, on modification date, and so on) they must all be true for a clip to be processed unless you check the 'OR' box, in which case just one of the OR terms needs to match. Check the 'NOT' back to invert that condition (ie. match all clips for which the condition is not true).

Just as with watch folders you should take care that you don't specify actions that will be performed in an endless loop. For example, if you have a server-triggered action based on clips whose status is "To be converted" you should update the status to something different such as "Completed" on successful completion in the post-processing stage. By default, the worker will only process each clip once per session but you can override this if required by checking the "allow polling" option.

Each clip that matches the server query will be added as a separate task into the task queue (as there could be a large number of files that each requires a lengthy transcode). If you perform an operation such as exporting an XML file on the Conversions tab it will therefore export one clip at a time – see the section below on 'Batch operation' if you want to export all the matching clips in one go.

Specifying a root folder

When processing files it's often a good idea to specify a 'root' folder. The file path is then calculated as being relative to this root folder which can result in shorter file paths. For example, if you have an original file in /Volumes/Media/Online Media/Project/File1.mov and want to build a proxy of that file in /Volumes/Media/Proxies then you should specify /Volumes/Media/Online Media as the root folder or the proxy will be /Volumes/Media/Proxies/Volumes/Media/Online Media/Project/File1.mov rather than /Volumes/Media/Proxies/Project/File1.mov.

If you use a watch folder files are taken relative to the watch folder itself but when dragging files on the worker icon you need to specify a root unless you want the entire original file path to be used. For server-triggered actions the path is taken relative to the root specified by any applicable path mapping. (To specify a root even though the original file path is already correct, specify a dummy path mapping from the root to itself.)

Job definitions

The job definition panel defines what actions are performed on files in the watch folder. You should give each job a short descriptive name so you know what the job does.

Any or all of the following actions can be configured, which will be performed in the given order:

1. The file is "imported" (see Pre-Processing section above)

2. Any movie conversions that you specify are applied. These can include creating CatDV previews or exporting the movie to one or more other locations and formats, at different resolutions. You can define your own movie conversion presets which are saved globally so you can apply them to different watch actions.

3. After the movie conversion(s) have all completed without error, which may take some time, post-processing can be applied to the original file in the watch folder. These include copying or moving the file – or **deleting** it. Take care when using this option and make sure it is what you want in your workflow! As well as specifying the destination directory you can specify how much if any of the path is preserved for those cases where you have subdirectories in the watch folder.

4. You can also export the clip(s) as an XML file and execute arbitrary operating system commands (see below). These can be combined so that the external command can read the value of clip fields from the XML file, and then update any desired clip fields as a result of its processing.

5. Finally, the clip(s) are published to the CatDV server so they are visible to users via the CatDV client application. You can publish all clips to the same catalog, or leave the catalog name blank and have a new catalog created each time. Normally, publish operations are 'batched' (using the Publish Interval setting) so one catalog might contain several files when it is published. If you converted the movie to a different format or automatically moved the file out from the watch folder you can choose to update the media reference in the clip to point to the new location.

6. Additionally, if you specify the file name for an HTML index file that file is automatically updated to include details of all the clips in the current catalog. You can also send an email notification to inform users that a new file is available.

Executing commands

As part of a job you can run an arbitrary operating system command invoked via a command line interface. Such commands might include copying a file by FTP, using an external transcoding engine such as FFmpeg or Telestream Episode, or running a batch command or shell script, and provides a very powerful way of extending the capabilities of the worker.

For example, under Mac OS X you could write

```
/usr/bin/ftp –u ftp://user:password@host/public/upload/ $l
```

to upload all the converted movie files to the /public/upload directory on 'host' via ftp. The 'curl' command gives even more options for copying files to an external server (including via https, sftp, etc.)

Up to three external commands can be performed, at different stages in the processing. The first command is intended for external transcoding or other pre-processing, the second command is intended for interaction with external systems via XML and supports updating clip fields as a result, while the final command could be used for FTPing completed files or performing notification that the action has completed.

To pass parameters to the external command you can either export the clip(s) being processed as an XML file (using an optional XSLT file to translate the file to another format from CatDV's default XML format) or pass them via the command line using a variable expression. To update clip fields as a result of processing the command check the 'parse output' option and write updates to the standard output of the process using fields ids such as NT for notes, one per line. For example:

```
INDEX=0                            -- if we are processing multiple clips specify which one
@STS=Complete                      -- set the clip Status field
@U5+=Processed by my command       -- append to the User 5 field
```

See below for a list of variables you can use use on the command line to refer to the original file being processed, the file(s) it was converted to, and more.

Variable expressions

When executing commands (and also when copying files, sending emails etc.) you can use the following variables to refer to the current file being processed and other special values.

$i	Full path of the input file (after preprocessing) (eg. "/Media/Original/file1.mov")
$f	The filename part only of the input file (eg. "file1.mov")
$g	The base part of the input filename after preprocessing (eg. "file1")
$G	The base part of the original filename before preprocessing (ie. before renaming)
$e	The extension of the input (eg. ".mov")
$E	The file extension, converted to upper case and omitting the period (eg. "MOV")
$p	Full path of the input file's parent directory (eg. "/Media/Original")
$P	The path of the input file's parent directory relative to the root (eg. "/Original")
$q	Name of input file's parent directory (eg. "Original")
$Q	Name of input file's grandparent directory (eg. "Media")
$R	Name of input file's great-grandparent directory, if any
$w	The file root directory that paths are relative to (eg. the watch folder)
$r	The original input file relative to the root directory
$h	The first path component after the root, if any
$b	Full path of the original input file that triggered the action (before preprocessing)
$B	Name of the xml batch file (if this job was triggered from a batch file)
$I	The existing server clip id (if this job was triggered by a server query)
$o	Full path of the converted movie (first movie only if multiple movies were created)
$z	Full path of the updated file that the clip will refer to, if different from $i
$c	A list of all the converted movies concatenated into one string
$l	A list of all the converted movies as separate arguments
$L	A list of all the files being added by a Folder Scan action as separate arguments
$K	A list of all the files being added by a Folder Scan action concatenated together
$a	Name of catalog the clips will be published to
$t	Tape name (if known)
$s	Starting timecode of the input file (if it's a supported media file containing timecode)
$d	Duration of the input file (if it's a supported media file) in timecode format
$n	Number of scenes that were detected (if scene detection was enabled)
$v	A summary of the format of the input file (if it's a supported media file)
$u	A long Unix timestamp that could be used to create a 'unique' filename
$U	A unique name dervived from the current timestamp, shown as letters and digits
$D	Current date as yyyymmdd, eg. 20081231
$M	Current year and month as yyyy mm, eg. 2009 01

$T	Current time as hhmmss, eg. 235959
$S	Task summary (available when sending email notification)
$j	The job name
$k	The task id
$m	The name of the importer being used
$x	The standard output of executing the previous command
$y	The error output of executing the previous command
${id}	Value of arbitrary clip fields, eg. ${U12} is User12, ${NT} is the Notes field
$1..9	Separate parts of the original filename, delimited by space or punctuation characters
$_d	The current day of month ($_D without leading zero), eg. 31
$_m	The current month number ($_M without leading zero), eg. 12
$_y	The current year ($_Y for last two digits of year only), eg. 2011
$_I	The current time as an ISO-formatted date (eg. "2015-07-21T11:45:17+00:00")
$_i	The current time formatted for user-defined date fields (eg. "2015-07-21 11:46:12")
$*	Special file naming rules in certain contexts (eg. next consecutively numbered file)
$_?	The matching filename in 'Test File Existence' steps with a wildcard
$\n	Include a literal new line character (also $\r for carriage return, $\t for tab, etc.)
$$	Include a literal '$' in the command.

Note that it is possible for a single input file to result in zero, one or many converted movies. First, you can define more than one movie conversion (for example, to create versions at different resolutions). Secondly, you can perform automatic scene detection and specify that each scene is converted to a separate file. Depending on what arguments the command you are executing expects you might therefore want to use either the $o, $c or $l variable.

Not all the variables will be available at all times. For example, $x is only available after executing command 1, clip fields are only available for server-based tasks or after importing a movie, and so on. This also means you can't use clip fields or the tape name when naming a catalog because the catalog is opened before the clip is imported.

To determine the id of clip fields you can select "Show attribute IDs" on the CatDV client and look at the identifier that appears in parentheses when choosing a field (for example when performing a query or customising a view). For example, '${CREF}' for clip id, '${ID2}' for the import date, '${U1}' for user field 1, '${@Album}' or '${@FNumber}' for a named metadata field when importing an MP3 or Exif file.

In the case of timecode values such as ${I1} for In point or ${O2} for Out2 you can provide a format modifier such as ${I1:t} to use the time as decimal seconds rather than a formatted timecode value, or ${I1:f} to display the number of frames. This is sometimes useful when passing command line arguments to a tool such as ffmpeg. In the case of timestamps such as ${RD1} or ${DT1} etc. for camera record date you can use ${RD1:s} to display a Unix timestamp (time in seconds since 1 Jan 1970), ${RD1:iso} to display in ISO format, and ${RD1:ago} to display how many seconds in the past from now the time is.

You can use variables in the following places:

- When specifying the subject, body or recipient of an email notification
- When setting or modifying a clip field such as Status or Notes
- When executing a command via the command line
- When naming the catalog to publish clips to on the server

- When specifying the path to copy or move a file to or specifying the destination of a movie conversion (if you choose ‘Expand variables’ as the file naming option)

When a variable contains a file path (separate components delimited with / on Mac or Unix, or \ on Windows) you can use array syntax to extract parts of the file path. You can count positions starting from either the left or right of the file path, they count down from 0 at the end or up from 1 from the beginning, so that $i[0] is the file name part of the input file (the same as $f) while $i[-1] is the parent directory (same as $q), and $i[1] is the first component of the path. You can also have ranges such as $p[1..2] or $p[-2..0].

Regular Expressions

A large number of useful variables that can be used for tasks such as constructing filenames are provided by default. For even greater flexibility, it is possible to modify the variable’s value using one or more regular expression substitution patterns. Regular expressions are commonly used in Unix and other scripting environments and provide a lot of flexibility, at the cost of a rather arcane syntax.

To use a regular expression substitution you should follow the variable with curly braces. The syntax is $*variable* ‘{‘ ‘s’ ‘/’ *search-pattern* ‘/’ *replacement-pattern* ‘/’ ‘}’. For example, if $x has the value “abcde” then $x{s/b/x} would be “axcde”.

The regular expression features provided are similar to those provided by Perl. Only the ‘s’ command is supported, but multiple commands can be included within the braces, eg. ${U1}{s/c/x/s/d/y/} to replace c with x and d with y in user field 1.

Any delimiter can be used after the ‘s’ instead of ‘/’, and the final delimiter before the ‘}’ may be omitted. The command may be followed by ‘g’ to apply the substitution globally to all occurrences (by default only the first occurrence is replaced) or ‘i’ to perform a case-insensitive match (by default it is case sensitive).

Within the replacement pattern any sub-elements of the search pattern enclosed within (round) parentheses can be included as ‘$1’ for the first sub-element, ‘$2’ for the second, and so on, so that $a{s/\D(\d*).*/$1/} would extract the first group of digits from $a. Other worker node variables can be included in the regular expression command, either within the search or replacement pattern. For example, $f{s/$_y//} would give you the value of $f with any occurrence of the variable $_y removed.

As a more complete example, let’s say we want to pick the first part of the parent folder name $p. Let’s consider how is the following statement, which looks like someone sneezed on the screen, processed: `$p{s,^.*/(\S+)[^/]*$,$1,}`

The curly braces mean take $p and modify it. What modification? The character after s is comma, so that's the delimiter and we're replacing *XXX* with *YYY*, whatever *XXX* and *YYY* may be: $p{s,*XXX*,*YYY*,}

XXX, the pattern we're searching for, is `^.*/(\S+)[^/]*$` which can be broken down as follows:

^ = start of string/start of line

$ = end of string/end of line

. = any character

* = previous pattern repeated any number (0 or more) times

/ = literal forward slash (if we're processing file paths on a Mac, use \\ instead if on Windows)

\S = any non-space character (\s = any white space character)

+ = previous pattern should occur 1 or more times

[..] = a list or range of characters to match

[^..] = negate the set of characters, ie. match any character not in the list

(..) = parentheses surround the part we're interested in

[^/]* = any combination of characters other than forward slash.

Combining all the above, this means take the variable $p, then try to match anything from the start (^.*), followed by a slash (/), followed by one or more non-space characters (\S+), followed by any number of characters excluding / ([^/]*), all the way up to the end of the string ($). All of this is replaced by $1.

We have to insist on there being a number of non-/ characters up to the end of the line after the text we're looking for as that's how we make sure we're picking up characters after the *last* forward slash, rather the *first* one.

The parentheses round the \S+ mean that we can refer to the characters matched by that expression later. That's what the $1 in the substitution pattern refers to, ie. we're matching the whole string and replacing it with just those non-space characters after the final /.

You can look up documentation for the 'java.util.regex.Pattern' class online for full details of the regular expression syntax that is supported, or press the 'Help' button in the watch item editor to see some examples. There are also a number of online tutorials and guides that describe regular expressions.

How file naming works

Many operations in the Worker involve creating new files (copies of existing files or rendered converted files) or renaming existing files, so various mechanisms are available to choose both the destination name and the folder structure.

A key concept is that of "relative paths", ie. the remainder of the file path relative to a particular root directory. When an action is triggered by a watch folder the watch folder itself is used as the root folder, but when the action is triggered by drag and drop or by a server trigger you can explicitly specify a root folder.

If you choose "Use filename only" the folder hierarchy is ignored and files are placed directly in the destination folder, whereas if you choose "Preserve entire path (from root)" the folder hierarchy relative to the root folder is maintained within the destination folder.

If you choose "Number files consecutively" or "Use current timestamp" then files are placed directly in the destination folder and the name is either an incrementing number (the next available number which isn't already in use in the destination folder) or based on the current date and time.

"Expand variables" gives you the most flexibility when naming files, and most of the other options can be implemented with variables, although it can sometimes be a bit tricky defining exactly what is required. When the action is "Rename files" the pattern you specify defines the filename only (as the file is being renamed, not moved to another folder) so you might enter something like "$Q_$f" as the destination. When copying or moving or converting a file, however, you need to specify both the destination folder and the filename, for example "/Media/Converted/$P/$D_$g.mp4".

When converting or exporting a file the correct filename extension (such as ".mov") is normally added automatically depending on the format being exported, unless you check the "exact name" option.

As there is so much flexibility in how files can be named (and in functionality that can be programmed with the worker more generally) you should always test your Worker scripts carefully with some simple test files before relying on a workflow. In particular, you should make sure that your worker configuration matches the media search paths configured in the CatDV client application preferences (and in the Live HTML/Web Publisher configuration if you are using the web interface) and that you have a thorough understanding of CatDV's concept of file paths *relative* to a particular root directory.

This complexity is necessary because no two workflows or installations are the same. Although it is usually possible and can be very tempting not to specify a root folder (so the entire file path is treated as if it was a relative path relative to the file system root "/") you should always try to consider what the logical root folder for a particular operation is because of the benefits this brings:

- Flexibility – if you add a new disk or need to move a folder it is much easier to reconfigure the system if you use relative rather than absolute paths. It is also possible to specify different actions or different search paths for different root folders.
- Efficiency – relative file paths are shorter, and in the case of the CatDV client searching for files using a search path vastly fewer file locations need to be checked if you correctly specify the root folder
- Security – rather than giving users access to the entire file system you can restrict it to particular folders and don't need to expose complete file paths (for example through the web interface)

Movie conversions

The main reason why the worker node implements a queue of tasks is of course because re-compressing video from one format to another tends to be a very time consuming process. You define the compression you require in the movie conversion editor.

First, you specify the destination for the converted movie. If you are creating CatDV previews then these are named automatically (based on the tape name and timecode value of the clip for tape-based previews, and based on the relative path from the root for path-based previews) and all you need to do is specify the CatDV preview root directory. Otherwise, you can specify any output directory you want and how you want the output file name to be derived given the input filename.

Then you specify the movie format itself by creating a new movie conversion preset or selecting one you defined earlier. A conversion preset can be used by several different task definitions to ensure they all create movies in a consistent format.

When defining a movie conversion preset you need to specify the exporter (such as QuickTime movie, MPEG-4, AVI etc.) and settings particular to that exporter, as well as general options such as the frame size and whether to add burnt in text such as a copyright message. You can either choose the exporter and customise the setting yourself or use one of the built-in settings. The original CatDV preview presets are available, optimised for different factors like speed of compression, but note that these are designed for very small file sizes and generally implement low frame rates of 8 or 12 fps so the OfflineRT, MPEG4 (Quality) and MPEG4 (Normal) settings are normally preferable. Or you can fully customise the movie export setting, using any QuickTime exporter, codec, frame rate and quality settings.

You can resample the movie to a smaller frame size by entering the desired bounds and whether to preserve the original aspect ratio by making the movie smaller than those bounds. If you leave these boxes blank it will use the original movie size (for "exported" movies) or assume the preferred size of

320x240 for CatDV previews. If you are creating a QuickTime .mov file the worker node will automatically add a QuickTime timecode track, which you can make visible or not as required.

After defining the compression settings you should enter a short descriptive name and save it as a conversion preset so you can reuse it when defining other watch actions.

As well as using the default QuickTime exporter you can also use the Xuggle exporter (if installed) and the DSJ DirectShow exporter (when running under Windows). Different exporters will have different capabilities and support different codecs but the basic approach for configuring them is the same in each case.

Exporting stills

As well as exporting movies, you can also define a conversion that will extract one or more JPEG still images of the media file.

To export a single still image from each clip you can choose to export either the first frame, the poster frame or the first thumbnail of that clip. There are also options that let you export all the thumbnails of a clip, all the event markers, or all event markers of a particular category. When exporting a single image the name is determined as specified in the file name dropdown, but when exporting multiple images the name is determined automatically by appending an incrementing number to the clip name. When choosing images of a particular marker category the name of that marker category is included in the filename. If you prefix the specified marker category with '*' then the name of the event marker is also included. (For example, if you choose to export markers of category "VIC" the resulting files will have names such as <clipname>-VICnn.jpg, whereas if you specify "*VIC" the files would be called <clipname>-VICnn-<eventname>.jpg). The "HTML Thumbnails" option exports all the thumbnails of a clip using the special naming scheme that CatDV uses when exporting clips as HTML or XML.

The images are always exported as JPEG files but you can define the maximum size (the image will be scaled so it doesn't exceed the specified size), the compression quality, and whether to preserve any Exif data from the media file.

Batch operation

If jobs are submitted via an xml batch file or as a result of a server query this will result in a separate task in the queue for each clip or file being processed. Actions defined on the Conversions and Post-Processing tabs, such as renaming the file or executing commands, will apply to each clip in turn, which may not be what you want if you want to process all the clips as a single batch. The following operations apply to an entire batch of clips in one go:

- You can export all the clips matching a server query into one HTML or XML file using the Batch Export section on the Parameters tab. This export is done once, before the individual files are processed. This applies to server queries only.
- To execute a command just once on completion of the batch (perhaps to submit all the clips to an external system in one go) use the special "Execute command 4" field on the Post-Processing tab. This applies to both server queries and batch file imports.
- Write a shell script to invoke the worker command line interface (see below).

The following special variables are available on batch completion and when naming the XML or HTML file to use:

$I	comma-separated list of server clip IDs (for server queries)
$B	the batch file name (for xml batch jobs)

$n total number of tasks processed in this batch
$m of this total, how many completed successfully
$c all the files processed by the batch (concatenated as one value)
$l all the files processed by the batch (as a list of separate values)

Also $j, $u, $U, $D, $M, $T, $_d, $_m, $_y have the same value as during normal task execution.

If you need to update the clips on the server during the batch completion phase you can do so using the worker command line by passing in the clip ID list, for example:

```
catdv -query ID:65:$I -set Status=Complete
```

You can monitor the status of a batch by looking at the batch id column in the task list and the batch details section on the details page for an individual task.

Batch vs individual operations

Most watch actions create a separate task for each clip or file being processed. This is suitable for actions which involve doing a full file import (full media file analysis and thumbnail extraction) or transcoding and where each task can take a signficant amount of time.

In some situations it is more efficient to operate on a large group of clips as a single task however, though in these cases you won’t be able to perform operations such as a transcode on each file. The following batch operations are available:

- Setting up a file watch folder with the “Folder Sync (Quick)” importer. Any new files that appear in the watch folder are quickly added to the catalog in one go, without doing any per-file analysis.
- Importing a CatDV XML file in immediate mode (see below) using the “CatDV XML File (Quick)” importer. This will quickly create clips using the data provided in the XML file without doing any per-file analysis.
- Importing a CatDV XML file in immediate mode (see below) using the “CatDV XML File” importer. This will do a full import of all the files referred to by the XML file in one task, so care should be taken with this option as the whole operation will fail if one file has an error and you also don’t get the benefit of scheduling different tasks across multiple processors if the import takes a long time.
- Performing a “Bulk Query”. This will perform a server query to find clips to process but rather than creating a separate task for each clip will process all the clips in one task. It is suitable for doing a bulk update of clips and then publishing changes back to the server but where no transcoding or media file analysis is required.

Offline workflows: batch import and reanalyse media

Occasionally it is necessary to create an asset record before the media for that asset is available. The Worker Node has new features to enable this way of working.

First, you can use a CatDV XML batch file to create new clip records without media being online. CatDV XML files can describe one or more clips including the clip name, media path, log notes, event markers, and technical metadata. (To see some example files you can export selected clips from CatDV Pro as a CatDV XML File.)

There are two distinct ways to import a CatDV XML file into the worker, batched and immediate. In both cases you define a watch folder where xml files will be placed.

In batched mode your watch action specifies that the job is triggered by a batch file and a separate task is added to the queue for each media file listed in the batch file. Batched mode is suitable if you want to build proxies for each media file but only works if the media files are online.

By contrast, in immediate mode the entire xml batch file is processed as a single task but one that can result in many clips being published to the server. Define a normal file triggered watch action but choose 'CatDV XML File' from the importer drop down. Ideally you should avoid doing lengthy transcodes when importing xml files in immediate mode but the advantage is that the files don't need to exist yet. If the file doesn't exist an offline clip is created instead, using the metadata specified in the xml file.

Secondly, the Worker Node now has the ability to reanalyse media. Just as with the Reanalyse Media command in CatDV Pro, this means analyse a media file to read technical metadata such as timecode, duration, and video format from the file as if it was being imported normally but instead of creating a new clip at that point update the metadata in an existing clip record, while preserving any user-entered metadata such as log notes.

There are two ways to reanalyse media, depending on what event you want to use to trigger the analysis.

You can define a normal watch folder so when a media file appears in that folder it is analysed and imported. On the Publish tab you can check the "Update existing" checkbox and if there is an existing offline clip in the catalog being published to whose clip name or media path matches the file being imported then that clip will be updated, otherwise a new clip will be created.

Alternatively, you can trigger the reanalysis based on a server query, possibly combined with the "Process online files only" option, by checking the new "Re-analyse Media" checkbox on the Pre-processing tab.

Development mode

When writing worker scripts it is sometimes annoying to have to wait for the normal poll and settle period before a worker action is performed (for example, if you have a large volume you might only scan for new files every five minutes). Additionally, if you have a lot of files and tasks that complete very quickly because all they do is move a file or publish a clip to the server without transcoding, it can be difficult to see what is happening because they run through so quickly. Both of these problems can be resolved by switching on development mode under general settings. Poll and settle periods are reduced to 10 seconds or less, and you are prompted before each job starts.

Other common techniques for troubleshooting worker actions include:

- Double click a task that has completed in the task list (or press the Info button) to view a summary of the operations that were applied to it and any errors that occurred.
- When configuring watch actions, temporarily disable actions you are not interested in by unchecking the Enabled item
- If a watch action doesn't seem to be triggered when you expect it to be, click the Test button. This displays useful information such as whether the watch folder is readable or the server query returns any matching clips.
- Turn on development mode
- View the worker log file, which may contain additional details of error messages. There are two (or more) log files. "CatDV Worker Log" displays details of the main worker process,

including editing the configuration and the output of the file watcher and server monitor threads, whereas the actual work of transcoding and performing actions is done by one or more separate helper processes, whose output goes to “CatDV Helper Log [2,3,4 etc.]”. The View Log button displays the main worker log, whereas if hold down shift and click the button you will see the latest helper log. You can also access the log files directly in Home > Library > Logs on the Mac (hold down the Option/Alt key then click on the Go menu in the Finder to navigate to the Library folder), or in “C:\Documents and Settings*Username*\Application Data\Square Box\Logs” (or similar) under Windows.

Hints and tips

Here is a collection of top tips for developing efficient Worker Node scripts:

- Ensure that all watch actions reset the conditions as part of the task so that the same clips aren’t processed over and over again. For server queries that look for a clip of status “Do Some Action”, update the status to “Action Done” on completion. For a watch folder, try to implement this as a drop box where files are moved out of the drop box to a more permanent location on completion.

- Avoid file conditions that need to analyse a file to check its contents. Conditions that only check the filename extension are much more efficient.

- Avoid creating very large thumbnails (or large numbers of thumbnails) unless you need them

- If possible, avoid using the “Create metaclip based on MXF UMID” and “Avoid duplicates” options, as they require the entire catalog that you are publishing to to be opened up by the worker before the task is processed

- Test your worker scripts on small numbers of example files before letting it loose on an entire volume. Use development mode (accessible from the General tab) to begin with so you only run one task at a time and can see what’s going on. You can also uncheck the various checkboxes in the Processing panel to slow things down.

- Initially test your scripts with source and destination on your local machine (eg. your desktop) and get them right there first before trying a network volume to eliminate network and permissions issues.

- Avoid reducing the check interval for watch folders or server queries below 60s unless you need to. Rather than speeding things up it might actually slow the worker down. (If you want to speed things up while testing your scripts, turn on development mode as this temporarily reduces the check interval.)

- Avoid the “allow polling” option unless you’re sure you need it

- Avoid having multiple workers on different machines sharing the same workset file. Instead, use multiple processes on the same machine.

- If possible keep the workset file on a local drive rather than a shared network drive.

- Keep the frame size and bit rate of your proxies down. Although the resulting file size isn’t as small as H264 or MPEG-4, using Photo-JPEG (OfflineRT) gives good quality and is much faster to encode and decode.

- Don’t leave the log viewer window open when you’re not using it

- When troubleshooting, reduce the number of processors to 1 (on the License tab) and quit and restart the worker to start a new log file (as the worker log file can get very large when it's been running for a while).
- If you have a lot of watch actions, disable all but the script that is causing problems then restart the worker. As well as keeping the log file small it will make troubleshooting easier if you only consider one watch action at a time.
- When reporting problems always provide the relevant log files (but see previous point about restarting the worker to keeping the file size down). Use the Save Log Files button from the log viewer to compress the relevant log file(s) into a ZIP archive, and don't forget to include a screenshot or precise details of which task failed (filename, time of day, error message, etc.) so we know which part of the log file to look at.
- If a file fails to import and get processed the way you expect it to in the worker, try importing it manually into CatDV Pro as that will make it easier to see any errors that are reported and you can check that you have the appropriate codec to play the file.
- Although the worker is fully cross-platform between Mac and Windows, there is a better choice of QuickTime codecs available on the Mac, so this can make a good platform for the worker even if the rest of the system is predominantly PC-based.
- If you have multiple worker licenses, try reducing the number of concurrent processes to see what gives the best performance with your workflow. If you're doing a lot of CPU intensive transcodes then increasing the number of processes may help but if you are processing a lot of short jobs then you might be constrained by the speed of I/O and applying too many concurrent processes may actually slow it down.
- If you change a job definition you need to resubmit any tasks that have already been queued to pick up the new definition.

Command line interface

Bundled with the worker node software is a new command line interface that can be used to query the CatDV database to generate customised reports, or to automatically update the server for use in external scripts of various kinds. Unlike the regular Worker Node, where automated actions are triggered from *within* the CatDV system (by a clip status change on the server or by means of a watch folder), the command line interface is ideally suited for system integrators who need to update clips as a result of an externally triggered event.

To use the command line tool you need to access it from a Terminal window or DOS command prompt. It's not a double clickable application. Either copy the 'catdv' executable to a location on your path (eg. /usr/local/bin or C:\WINDOWS\SYSTEM32) or edit your PATH variable. Alternatively, you can specify the full path to the executable. Once you have done this, typing "catdv -help" will display a list of available commands.

The CatDV command line interface requires a valid Worker Node license on the machine to run, and automatically picks up configuration details such as the server IP address (as well as the license activation code) from the Worker configuration file on that machine. The following settings are read from the worker config file, so you need to use the Worker Node application to edit these parameters: license code, server IP address, CatDV user name and password used to log on to the server, number of user-defined fields, and date format.

Commands

The following commands are available:

catdv -help, catdv -helpfields, catdv -helpquery

> Display a summary of available commands, a list of fields and their identifiers, and information about defining clip queries.

catdv -version, catdv -config

> Display the version number of the application or a summary of the configuration information read from the CatDV Worker Node file.

catdv -status

> Establish a connection to the server and display statistics about the database.

catdv -getProperty *property*

> Return a server property. Valid property names include 'version', 'startupError', 'warningMessage', 'isEnterprise', 'isRegistered', 'regLicense', 'regExpires', 'catdv.allowAnonymous', 'catdv.useRecycleBin'

catdv -groups, catdv -users

> Display a list of all the production groups and users on the server, including their ids. (A format and output file can be specified if required, see below).

catdv [*query*] -catalogs (or -catalogsdetailed)

> Display a brief or detailed list of catalogs on the server. If *query* is omitted, all the catalogs are listed (and the clip count is the total number of clips in each catalog), whereas if a query is specified only those catalogs that contain or more matching clips are shown, and the clip count is the number of clips that match the query. (A format and output file can be specified if required, see below).

catdv [*query*] -distinct *field*

> Display all the distinct values of a grouping field (such as Bin or Status) that occur in the database (optionally a query may be specified).

catdv [*query*] -summarise *field*

> Similar to the -distinct command, this displays a list of all the distinct values of a field, together with a count of how many times that value occurs. If a query is specified, then only clips that match the query are considered. (A format and output file can be specified if required, see below).

catdv -thumbnail *id*

> Extract a specific thumbnail image from the database given the thumbnail id (typically obtained from the poster id clip field). Binary JPEG data is returned and the results will normally be redirected to a file.

catdv [-catalog *name* | -catalogid *id*] –insertclip *field*=*value* …

> Insert a new clip into the database. An existing catalog must be specified by id or name. Any number of clip fields can be specified, ideally clip name, type, timecode format, plus media path, in and out points should be specified if known.

catdv *query* -count

Perform a clip query and display a count of how many clips match.

catdv *query* [*options*] -print [*field,…*]

Perform a clip query and display the matching clips. A list of which clip fields to display can be specified, otherwise a default list consisting of clip id, clip name, tape name, in and out timecode, format summary, and media file path is shown. (A format and output file can be specified if required, see below).

catdv *query* -print1 *field*

Perform a clip query and display a single field from all the matching clips. (Unlike the -print command, which returns full details of all the matching clips from the server and then selects which fields to display, this command is optimised to only return the requested data.)

catdv *query* -deleteclips

Perform a clip query and delete the matching clips, without asking for confirmation (though depending on how the server is configured, clips may be moved to the recycle bin rather than being deleted immediately).

catdv *query* -set *field*=*value* …

Perform a clip query and then update one or more fields of all the matching clips before publishing changes back to the server. Use -set *field*+=*value* to append the new value on the end of the existing value for the clip.

Fields and query terms

To specify a particular clip field you can use its short identifier (eg. “NM1”), its display name (eg. “Status”) or its internal numeric identifier, if known (eg. 17).

To specify a query, one or more query terms need to be specified. By default, multiple terms are combined with a logical AND operator (ie. only those clips which match all the query terms are processed). The following query terms are supported:

-name *text*

Only return clips whose clip name includes the specified text

-anyfield *text*

Return clips where are any logging or media field includes the specified text. (Logging fields include name, notes, bin and user-defined fields, while media fields also include read-only media metadata fields such as Exif exposure information, ID3 album and artist tags, P2 camera metadata, and similar fields.)

-anylogfield *text*

Return clips where any logging field includes the specified text.

-groupid *id*

Only return clips from the specified production group (whose id was returned by the -groups command)

-mediafile *text*

Return clips whose media path contains the specified text

-clipid *id*

Return a specific clip given its id

-all

Return *all* clips in the database, if no other query terms are specified.

-catalogid *id*

Only return clips in the specified catalog. Use -catalogid 0 to display the contents of the recycle bin.

-catalog *name*

Specify the catalog by name

-query [*modifiers*]*field*=*value*

Return clips where the specified field exactly matches a particular value.

-query [*modifiers*]*field*:*op*:*value*

Specifiy a completely general query term, consisting of a field, comparison operator and value to compare against, together with optional modifiers. The following modifiers are recognised: % (make the comparison case sensitive, normally case is ignored), ^ (negate the query term, so only clips which *don't* satisfy the condition are returned), * (include this term in a list of terms combined with a logical OR operator, where at least one of the terms must be satisfied for a clip to be returned). Fields may be specified either by name or identifier, as elsewhere. The comparison operator is given by a number, such as 4 for 'is blank' or 35 for 'timecode between two values'. Type "catdv -helpquery" for a full list of comparison operators.

Options

A number of modifier options are available with many of the commands above:

-limit *number*

Only return the given number of clips at most. (By default, a limit of 50000 clips is applied. Specify 0 to turn off the built-in limit.)

-members

Include all the members inside a metaclip rather than just displaying one entry for the metaclip as a whole.

-format *fmt*

Specify the format and delimiter when printing results. Use 'tab' (the default) or 'tabNoHeader' for tab-separated text, 'csv' for comma-separated format, 'bar' to separate fields with a vertical bar character (|), and 'xml' to use an XML format.

-out *file*

By default, all results are printed to the console, unless -out is used to redirect the output to a specified file (depending on how you invoke the command you can also use shell redirection with the '>' operator of course).

-sort *field*[,…]

When printing results, sort the clips based on the specified field(s). These need not necessarily be in the list of fields being displayed. Precede the field name with '-' to reverse the sort order.

Examples

catdv -groups

Look up the identifier for a specific production group, for use in subsequent queries

catdv -groupid 12401 -summarise Status

Display a summary of how many clips there are with each status in the given production group

catdv -query U1=Smith -query Status=Approved -format xml > /tmp/smith.xml

Generate a report of approved clips relating to a particular customer

catdv -mediafile DSC01245.JPG -print ID,NM1,MF,P1 -sort -ID1

Display all clips whose media file matches the given name and print a summary, including the poster id. Sort them so the most recently imported file comes first.

catdv -thumbnail 189039 -out ~/Desktop/thumb01245.jpg

Extract the specified poster image to a file.

catdv -query "*U1=Interview" -query "*Bin=Interview" -print

Display clips where either User1 or Bin is "Interview". When typing certain punctuation characters (eg, *, $, ! etc.) be aware that they may have special meaning to the command interpreter shell and may need to be "escaped" by enclosing them in single or double quotes.

catdv -query ^STS:4: -count

Display a count of how many clips have a status which isn't blank (operator 4 means 'is blank').

catdv -query 108:2:/Volumes/Media -summarise 108

Display a count of how many clips there are in each folder below /Volumes/Media (108 is the numeric attribute id for 'media folder', operator 2 means 'starts with')

catdv -query U2:4: -query STS:4: -query MD1:22:,86400 -set U2=Peter -set Status=Overdue

Take all clips whose status and assignee (user field 2) are blank and which are more than 24 hours old (MD1 is modified date, operator 22 means 'older than', and 86400 is the number of seconds in one day) and assign them to Peter, marked as being overdue.

catdv -query ID:65:11965,11968,11979,11988,… -set ARCHV="Archived to tape B17"

Update the archive status of a specified list of clips (whose identifiers were determined previously) to indicate that they have been successfully archived to tape. (Operator 65 means 'is one of').

Configuring the worker via XML

The worker node configuration is stored in an xml file. Normally the configuration file is edited via the worker user interface but it is also possible to edit this file directly.

The configuration file has both common definitions, such as the IP address of the CatDV workgroup server to which clips are published, and one or more workset definitions which contain lists of watch folders and actions to perform on files that appear in them.

The configuration file is called worker.xml and is in the home directory of the user running the woker software (eg. /Users/joesmith or C:\Documents and Settings\joesmith).

In normal use the config editor will be used to edit the worker node configuration. There may be situations when it is useful to edit the config file manually however, so the file format is documented below.

For example, the following configuration file implements a Drop Box folder on the desktop where any movie files are converted to MPEG-4 files and are then deleted. Scene detection is applied (a separate MP4 file is create for each scene) and the clips are published to the “AutoPublish” catalog on the server. Any JPEG images in the drop box are also published to the server and moved to the Images folder.

```
<config>
    <server host="localhost:1099" reguser="Joe Smith" regcode="ZVAD-…-7VKC"
/>
    <format name=”My MPEG4 settings” exporter="mpg4"
settings="*mpg4++++++++++%01+++%04%B7sean+++%01+++%14++++++%01%9Evide+++%01++
+%0F+++++++%22sptl+++%01++++++++mp%3++%20++%03++++%20tprl+++%01 …"/>
    <workset lockfile="/Users/joesmith/workset.xml">
        <watch folder="/Users/joesmith/Desktop/Drop Box" recurse="true"
period="60" settle="300" fileOp="delete" pattern=”*.mov” regex=”false”>
          <job name="Custom mp4" publish="true" catalog="AutoPublish"
thumbSize="2" analyseOther="true" >
            <convert dest="/Users/joesmith/Desktop/MPEG4"
destName="parent" format=”My MPEG4 settings”/>
          </job>
        <watch folder="/Users/joesmith/Desktop/Drop Box" recurse="true"
period="60" settle="300" fileOp="delete" pattern="*.jpg">
            <job name="Publish jpg" publish="true" catalog="AutoPublish"
thumbSize="3" fileOp="move" fileDest="/Users/joesmith/Desktop/Images"/>
        </watch>
    </workset>
</config>
```

As this example illustrates, it is possible to configure the worker node to automatically delete files in the watch folder after they have been processed (as this may form a valid part of the workflow). Take great care therefore when configuring such workflows, and make sure the root of the watch folder is set correctly!

Support

Please contact your systems integrator, email support@squarebox.com, or visit our web site at http://www.squarebox.com if you have any questions or require technical support for this product.

Square Box Systems Ltd
Lake View House,
Tournament Fields,
Warwick,
CV34 6RG
United Kingdom