POD Translation by *pod2pdf*

ajf@afco.demon.co.uk

dra-backup

Table of Contents
dra-backup

DRA-BACKUP

dra-backup - Copies files from remote recording stations

VERSION

This documentation refers to dra-backup version 2.1

USAGE

dra-backup [-help] [any options from dra.conf]

OPTIONS

-help Displays brief synopsis of usage.
dra.conf Any setting from the dra.conf file may be passed to the script at runtime in the format:
-[variable name] [value]
Example:
-mountPoint
Full path to the location where the samba share should be mounted. This should be specified in the dra.conf file and left unchanged.
-group
When connecting to samba shares over the network, the client initiating the connection will need to specify under which group the connecting user‘s credentials are found. This is usually the local group on the computer hosting the samba share such as BL-MUS-[NAME] and should distinguish a local user account versus a network account under the ADS network directory.
-smbUser
The username that as acces to the specified share on the hosting computer.
-smbPass
The password for the above user. Note: if the password is given in a script, it will be readable.
-share
The fully-qualified domain name of the computer hosting the samba share and the name of the share. Format should be: [computer_name]/[share_name]. See above examples.

DESCRIPTION

This script syncs data from remote recording stations to a folder located on the host machine. Two syncs are performed. The first is a backup and all the data under the target directory on the remote machine is copied to a local folder on the host machine. The second sync is intended for processing and only certain project folders on the remote station are copied.
Remote stations are backed up by mounting a samba share and then executing an rsync command. The sharepoint is exported from the remote station and then mounted from the host server.
The presence of a folder called "access" under each specific project folder, indicates that the concert is ready to be transferred. This is the indicator that the project should be copied to the semester folder for additional processing by more scripts. During this stage permissions are set according to the setPermissions subroutine in the DRA.pm file.
If no options are specified when the command is run, the defaults in DRA_HOME/conf/dra.conf are used. Passing any of the variables listed in the dra.conf file as switches overrides the values given in the dra.conf file. See EXAMPLES for more information

DIAGNOSTICS

Upon successful execution, the application with exit with no output. Any fatal error will be written to standard output and the application ceases immediately.
The most common failures would be: 1) the presence of any blocking processes as listed in the dra.conf file, 2) an unavailable mount point on the client machine, 3) an unavailable share on the remote station, 4) incorrect credentials for accessing the samba share on the host computer, and 5) absence of the DRA_HOME environment variable.
Detailed information about other non-fatal errors will be found in the application log under the log directory.

CONFIGURATION AND ENVIRONMENT

Once properly configured with the dra.conf file, only a minimal number of options need to be specified such as different remote shares, groups, and/or users and passwords.
A default username and password can be specified in the dra.conf file and then overridden at execution time with the appropriate switch passed via the command line.

The DRA_HOME environment variable is required and can be set on a per-user basis or at execution time. For example:
DRA_HOME=/usr/local/audio dra-backup [options]

DEPENDENCIES

All required modules are found under DRA_HOME/lib and other assisting scripts are found under DRA_HOME/tools

AUTHOR

Adam Wead awead@indiana.edu, amsterdamos@gmail.com

LICENCE AND COPYRIGHT

POD Translation by *pod2pdf*

ajf@afco.demon.co.uk

dra-convert

Table of Contents
dra-convert

DRA-CONVERT

dra-convert - Converts preservation audio files to access format

VERSION

This documentation refers to dra-convert version 2.1

USAGE

dra-convert [-help] [dra.conf options] [project_directory]

REQUIRED ARGUMENTS

A **project directory** is required for the script to run. This must be the full path the directory. For example:
dra-convert /Volumes/audio/Concerts/2010_fall/20100923vab1234

OPTIONS

-help Displays brief synopsis of usage.
dra.conf Any setting from the dra.conf file may be passed to the script at runtime in the format:
-[variable name] [value]
Example:
dra-mdss -email joebob@indiana.edu [directory]
Will execute the script overriding the email address specified in dra.conf with joebob@indiana.edu. Generally, this script is called during the dra-transfer script, so passing options to this script is not needed. However, one may call this script individually if needed.

DESCRIPTION

The dra-convert script locates all the wave files found in the project directory and converts them from 24/96 to 16/44 resolution. During this process, it also computes gain across all the files and applies the correct amount of normalization to each file.
Temporary files are written somewhere else other than the project directory so that once complete, the newly created access files will be copied over the to access directory of the project directory. If there is any failure along the way, temp files are removed and process can repeat itself until there is a successful outcome.
The script will attempt to convert all audio files until there is success. This is determined by the absence of an error log under the project‘s log directory and the presence of conversion log as well.

DIAGNOSTICS

Upon successful execution, the application with exit with no output. Any fatal error will be written to standard output and the application ceases immediately. Messages are written to the application‘s own log file found in the applications log directory, as well as a separate log file found in the project‘s log directory.
In the case of errors written to standard out, the error will appear on the command line shell or will be collected by the calling script, usually dra-transfer. The presence of any fatal error will cause the script to write an error log to the project log directory found in the access folder of the project. The presence of this file will mean the conversion will be repeated again if the script is called.

CONFIGURATION AND ENVIRONMENT

A properly configured dra.conf file should be sufficient for the dra-convert script to run.
The DRA_HOME environment variable is required and can be set on a per-user basis or at execution time.

DEPENDENCIES

This script requires the Apple program **afconvert** as well as the third party software program **normalize**. Afconvert should come with every OSX installation and normalize is available for free via GPL. Generally, it should be installed to /usr/local/bin but it can be installed anywhere so long as its location is given in dra.conf.

AUTHOR

Adam Wead awead@indiana.edu, amsterdamos@gmail.com
Travis Gregg

LICENCE AND COPYRIGHT

POD Translation by *pod2pdf*

ajf@afco.demon.co.uk

dra-delete

Table of Contents
dra-delete

DRA-DELETE

dra-delete - removes completed audio projects from their remote stations

VERSION

This documentation refers to dra-delete version 2.1

USAGE

dra-delete [-help] [dra.conf options]

OPTIONS

-help Displays brief synopsis of usage.
dra.conf Any setting from the dra.conf file may be passed to the script at runtime in the format:
-[variable name] [value]
Example:
-mountPoint
Full path to the location where the samba share should be mounted. This should be specified in the dra.conf file and left unchanged.
-group
When connecting to samba shares over the network, the client initiating the connection will need to specify under which group the connecting user‘s credentials are found. This is usually the local group on the computer hosting the samba share such as BL-MUS-[NAME] and should distinguish a local user account versus a network account under the ADS network directory.
-smbUser
The username that as acces to the specified share on the hosting computer.
-smbPass
The password for the above user. Note: if the password is given in a script, it will be readable.
-share
The fully-qualified domain name of the computer hosting the samba share and the name of the share. Format should be: [computer_name]/[share_name]. See above examples.

DESCRIPTION

Similar to the dra-backup script, dra-delete access remote recording stations except to remove complete project files.
The dra-delete script compares any project directories on a remote station to directories found locally on the server. If a project has finished processing without any errors, the script will remove the project from the remote station as well as from the backup folder on the local server.
A project is considered finished when all four project logs exist: copied log, conversion log, burned log and mdss log; and there are no error logs.

DIAGNOSTICS

Upon successful execution, the application with exit with no output. Any fatal error will be written to standard output and the application ceases immediately.
The most common failures would be: 1) the presence of any blocking processes as listed in the dra.conf file, 2) an unavailable mount point on the client machine, 3) an unavailable share on the remote station, 4) incorrect credentials for accessing the samba share on the host computer, and 5) absence of the DRA_HOME environment variable.
Detailed information about other non-fatal errors will be found in the application log under the log directory.

CONFIGURATION AND ENVIRONMENT

Once properly configured with the dra.conf file, only a minimal number of options need to be specified such as different remote shares, groups, and/or users and passwords.
A default username and password can be specified in the dra.conf file and then overridden at execution time with the appropriate switch passed via the command line.
The DRA_HOME environment variable is required and can be set on a per-user basis or at execution time. For example:
DRA_HOME=/usr/local/audio dra-delete [options]

DEPENDENCIES

All required modules are found under DRA_HOME/lib and other assisting scripts are found under DRA_HOME/tools

AUTHOR

Adam Wead awead@indiana.edu, amsterdamos@gmail.com

LICENCE AND COPYRIGHT

POD Translation by *pod2pdf*

ajf@afco.demon.co.uk

dra-mdss

Table of Contents
dra-mdss

DRA-MDSS

dra-mdss - Transfers projects to IU‘s mass data storage facility

VERSION

This documentation refers to dra-mdss version 2.1

USAGE

dra-mdss [-help] [-f] [dra.conf options] [project_directory]

REQUIRED ARGUMENTS

A **project directory** is required for the script to run. This must be the full path the directory. For example:
dra-mdss /Volumes/audio/Concerts/2010_fall/20100923vab1234

OPTIONS

-help Displays brief synopsis of usage.
-f Force flag. The project will be transferred to mdss regardless of any errors or if the concert was successfully sent at an earlier time.
dra.conf Any setting from the dra.conf file may be passed to the script at runtime in the format:
-[variable name] [value]
Example:
dra-mdss -email joebob@indiana.edu [directory]
Will execute the script overriding the email address specified in dra.conf with joebob@indiana.edu

DESCRIPTION

dra-mdss will take a project folder that had been successfully processed by the dra-transfer script and send it to the mdss storage facility for archiving. During this processes, files will be renamed and checksums will be computed and stored along with the original files to ensure accurate transfers in the future.
The script will continue to process a project directory until there is a successful outcome. Once the project has been successfully sent to mdss, any further executions on that particular project directory will have no effect. Using the force option, (-f) will override this and re-send the project to mdss.

DIAGNOSTICS

Upon successful execution, the dra-mdss with exit with no output. Any fatal error will be written to standard output and the application will exit immediately. Messages are written to the application‘s own log file found in the application log directory, as well as a separate log file found in the project‘s log directory in the access folder.
In the case of errors written to standard out, the error will appear on the command line shell or will be collected by the calling script, usually dra-run. The presence of any fatal error will cause the script to write an error log to the project log directory found in the access folder of the project. The presence of this file will mean dra-mdss will run again when called either directly or by dra-run.

CONFIGURATION AND ENVIRONMENT

A properly configured dra.conf file should be sufficient for dra-mdss script to run.
The DRA_HOME environment variable is required and can be set on a per-user basis or at execution time.

DEPENDENCIES

HTAR and HSI are required to send the files to MDSS. These applications are available through IU‘s mass datastore service. See http://kb.iu.edu/data/auvo.html for the latest client builds available.
You will also need to create kerberos keytab files to enable automate access to the mdss account. For more information on creating keytab files, see http://kb.iu.edu/data/avdb.html.

AUTHOR

Adam Wead awead@indiana.edu, amsterdamos@gmail.com
Travis Gregg

LICENCE AND COPYRIGHT

POD Translation by *pod2pdf*

ajf@afco.demon.co.uk

dra-notify

Table of Contents
dra-notify

DRA-NOTIFY

dra-notify - Executes a script and sends out an email notification of the results

VERSION

This documentation refers to dra-notify version 2.1

USAGE

Usage: dra-notify [-help] [script]

REQUIRED ARGUMENTS

A **script** or executable file is required for dra-notify to run. Use the path relative to DRA_HOME
Examples:
To execute dra-run and have dra-notify send you the results:
dra-notify bin/dra-run
To run the Auer Hall script:
dra-notify -name "Auer Hall" scripts/auerHall.sh

OPTIONS

-help Displays brief synopsis of usage.
-name Allows you to specify a name for the script in the subject line of the email. Otherwise, the path to the script will be used.
-email Overrides the email setting in dra.conf.
-sender Overrides the sender setting in dra.conf.

DESCRIPTION

dra-notify is a simple wrapper command that executes any script and emails the results.

DIAGNOSTICS

dra-notify only reports the success or failure of the script it is calling.

CONFIGURATION AND ENVIRONMENT

A properly configured dra.conf and the DRA_HOME environment variable are required.

AUTHOR

Adam Wead awead@indiana.edu, amsterdamos@gmail.com

LICENCE AND COPYRIGHT

POD Translation by *pod2pdf*

ajf@afco.demon.co.uk

dra-run

Table of Contents
dra-run

DRA-RUN

dra-run - Executes dra-transfer, marc.pl and dra-mdss for a semester‘s projects

VERSION

This documentation refers to dra-run version 2.1

USAGE

dra-run [-help] [dra.conf options]

OPTIONS

-help Displays brief synopsis of usage.
dra.conf Any setting from the dra.conf file may be passed to the script at runtime in the format:
-[variable name] [value]
Example:
dra-run -email joebob@indiana.edu [directory]
Will execute the script overriding the email address specified in dra.conf with joebob@indiana.edu

DESCRIPTION

dra-run executes a series of scripts and actions on each project directory found within a given folder. These include: dra-transfer, marc.pl, email notifications to Jira, and dra-mdss. After every project directory is processed, the accumulated marc records for successful actions are emailed for loading into IUCAT.

ERROR HANDLING

dra-run will interpret any output, whether from STDOUT or STDERR, as an indication that something went wrong. If any of the dra scripts outputs something, it generally means that something did not go as it should have. Conversely, if there is no output from either STDOUT or STDERR, then the script was successful.
dra-run also behaves in the same manner, so that if there is no output from any of the collective scripts that run, indicating that everything ran successfully, dra-run returns no output either.

DIAGNOSTICS

dra-run only collects and reports on the outcomes of other scripts. Errors regarding the outcomes of those scripts will be found in the respective project log folder. There is an application log for dra-run to report progress and any non-fatal errors.
dra-run would only fail for environmental reasons such as incorrect permissions or other problems interfering with the application. Otherwise, the dra-run reports explicitly on the success or failure of a given project.

CONFIGURATION AND ENVIRONMENT

A properly configured dra.conf file should be sufficient for dra-run to work.
The DRA_HOME environment variable is required and can be set on a per-user basis or at execution time.

DEPENDENCIES

The marc.pl script under the DRA_HOME/tools folder is required for marc record creation.

AUTHOR

Adam Wead awead@indiana.edu, amsterdamos@gmail.com

LICENCE AND COPYRIGHT

POD Translation by *pod2pdf*

ajf@afco.demon.co.uk

dra-transfer

Table of Contents
dra-transfer

DRA-TRANSFER

dra-transfer - Checks, converts and transfers a single project to Variations

VERSION

This documentation refers to dra-transfer version 2.1

USAGE

dra-transfer [-help] [-r -f] [dra.conf options] [project directory]

REQUIRED ARGUMENTS

A **project directory** is required for the script to run. This must be the full path the directory. For example:
dra-convert /Volumes/audio/Concerts/2010_fall/20100923vab1234

OPTIONS

-help Displays brief synopsis of usage.
-r Retransfer switch. Transfers a project even if it has already been successfully sent before.
-f Force flag. Performs a force transfer where project directory checks are skipped and the concert is sent as-is.
dra.conf Any setting from the dra.conf file may be passed to the script at runtime in the format:
-[variable name] [value]
Example:
dra-run -email joebob@indiana.edu [directory]
Will execute the script overriding the email address specified in dra.conf with joebob@indiana.edu

DESCRIPTION

dra-transfer checks the consistency of a given project directory, such as filenames and contents. It then converts the audio files using dra-convert, and sends the files to Variations.
The presence of a directory named "video" in the project indicates that the project has video files residing on different computer. The video processing computer is responsible for sending a text listing of files from the project directory to the project log directory on the dra server. If dra-transfer runs, and detects a video folder but no corresponding text file with the file names, the script will abort and not continue until this text file is present.
In preparation for dra-mdss and reporting to Variations, dra-transfer will create a data file and text file with the new names of the files in the directory. However, the filename will not change until dra-mdss runs using the data file created by dra-transfer. Successfully complete projects will also have some of the contents copied to the burn folder for CD creation.
Once a project is successfully converted and transferred, any further action on it will have no effect unless the force (-f) or resend (-r) flags are used. Any errors are written to an error log in the project‘s log directory. The presence of this log will cause dra-transfer to attempt the transfer the project the next time it is called.

DIAGNOSTICS

Upon successful execution, the dra-transfer with exit with no output. Any fatal error will be written to standard output and the application will exit immediately. Messages are written to the application‘s own log file found in the application log directory, as well as a separate log file found in the project‘s log directory in the access folder.
In the case of errors written to standard out, the error will appear on the command line shell or will be collected by the calling script, usually dra-run. The presence of any fatal error will cause the script to write an error log to the project log directory found in the access folder of the project. The presence of this file will mean dra-transfer will run again when called either directly or by dra-run.
In cases where video is involved, the video processing computer will not send any files to the dra server or MDSS until this script has completed sending all audio files to Variations and MDSS successfully. For more information, refer to the documentation pages for dra-video.

CONFIGURATION AND ENVIRONMENT

A properly configured dra.conf file should be sufficient for dra-transfer to work.
The DRA_HOME environment variable is required and can be set on a per-user basis or at execution time.

DEPENDENCIES

The dra-convert script is required.

AUTHOR

Adam Wead awead@indiana.edu, amsterdamos@gmail.com

LICENCE AND COPYRIGHT

POD Translation by *pod2pdf*

ajf@afco.demon.co.uk

dra-video-run

Table of Contents
dra-video-run

DRA-VIDEO-RUN

dra-video-run - Executes dra-video on multiple project directories

VERSION

This documentation refers to dra-video version 2.1

USAGE

dra-video-run [-help] [-f] [video.conf options]

REQUIRED ARGUMENTS

None

OPTIONS

-help Displays brief synopsis of usage.
video.conf Any setting from the video.conf file may be passed to the script at runtime in the format:
-[variable name] [value]
Example:
dra-video-run -email joebob@indiana.edu [directory]
Will execute the script overriding the email address specified in video.conf with joebob@indiana.edu

DESCRIPTION

This script gathers a list of project directories under one or more parent directories listed in the video.conf file. dra-video-run then calls dra-video on each of these projects directories individually and reports on the outcome of each one at the completion of the script.
The script has its audio counterpart in dra-run, except that dra-video-run is simpler and no additional procedures are performed other than executing dra-video on each project directory. Like dra-run, dra-notify is used in conjunction with dra-video-run to email the results of processed project directories to individuals specified in the dra.conf file.

DIAGNOSTICS

See the documentation for dra-video under this section.

CONFIGURATION AND ENVIRONMENT

See the documentation for dra-video under this section.

DEPENDENCIES

See the documentation for dra-video under this section.

AUTHOR

Adam Wead awead@indiana.edu, amsterdamos@gmail.com

LICENCE AND COPYRIGHT

POD Translation by *pod2pdf*

ajf@afco.demon.co.uk

dra-video

Table of Contents
dra-video

DRA-VIDEO

dra-video - Transfers video files to local server and MDSS

VERSION

This documentation refers to dra-video version 2.1

USAGE

dra-video [-help] [-f] [video.conf options] [project_directory]

REQUIRED ARGUMENTS

A **project directory** is required for the script to run. This must be the full path the directory. For example:
dra-video /Volumes/audio/Concerts/2010_fall/20100923vab1234

OPTIONS

-help Displays brief synopsis of usage.
-f Force flag. The project will be transferred regardless of any errors or if the concert was successfully sent at an earlier time.
video.conf Any setting from the video.conf file may be passed to the script at runtime in the format:
-[variable name] [value]
Example:
dra-video -email joebob@indiana.edu [directory]
Will execute the script overriding the email address specified in video.conf with joebob@indiana.edu

DESCRIPTION

This script works in conjunction with dra-transfer to coordinate transfers of video files to the main dra server and to MDSS. Whereas dra-transfer and dra-mdss are intended to be run from a single machine or server where audio data is "pulled" from different workstations, dra-video is designed to be run on multiple workstations where video data is "pushed" to both the main dra server and mdss.
Processing a project directory will take two executions of the dra-video script. On the first pass, the script will first check for properly named files and folders. If these are correct, it will next create a text listing of files and send them to the main dra server. A corresponding project directory will already exist on the main dra server with any relevant audio data. Once this file listing is copied to the server, the dra-transfer script will be able to complete its transfer of audio data to Variations and MDSS. See the documentation for dra-transfer and dra-mdss for more information.
On the second pass of the script, dra-video logs into the main dra server to determine if both dra-transfer and dra-mdss have completed successfully. Upon the successful complete of both those scripts, dra-video will transfer dvd image files to the main dra server, and then transfer all video files to MDSS.
Individual video files are stored via HSI, and dvd folders are archived via HTAR.
The last step of the script is to copy the video transfer log to the dra server for additional storage.

DIAGNOSTICS

Upon successful completion of the script, dra-video returns no output. Otherwise, errors are reported to STDOUT as well as an error log file in the project log directory. Additional information about the scripts overall progress is written to a script-level log file under the script log directory.
By default, this is the logs directory in the application‘s home directory.

CONFIGURATION AND ENVIRONMENT

A properly configured video.conf file should be sufficient for dra-video script to run. The format is the same as the dra.conf file and has many duplicated values. However, there is no shared information in either of the configuration files. So any relevant changes to dra.conf must also be duplicated in video.conf and vice versa.
The DRA_HOME environment variable is required and can be set on a per-user basis or at execution time.

DEPENDENCIES

HTAR and HSI are required to send the files to MDSS. These applications are available through IU‘s mass datastore service. See http://kb.iu.edu/data/auvo.html for the latest client builds available.
You will also need to create kerberos keytab files to enable automate access to the mdss account. For more information on creating keytab files, see http://kb.iu.edu/data/avdb.html.

AUTHOR

Adam Wead awead@indiana.edu, amsterdamos@gmail.com

LICENCE AND COPYRIGHT