

NAME

moneyworks – MoneyWorks command line modes

SYNOPSIS

moneyworks [options] [file-or-url]

On Mac OS X, the short form (moneyworks) is a symbolic link to the actual executable (/Applications/MoneyWorks\ Gold.app/Contents/MacOS/MoneyWorks\ Gold). On Windows, a separate command line client, moneyworks.exe is installed in C:\Program Files\ MoneyWorks Datacentre\. See notes for caveats on symbolic link usage.

DESCRIPTION

On Mac OS X, MoneyWorks Gold can be executed from the command line in either normal GUI mode with an optional file or URL parameter, or as a non-GUI command line tool. In the latter case, it is useful for automated access to local or remote MoneyWorks Gold databases.

On Windows, moneyworks.exe is a command-line-only tool (vs MoneyWorks Gold.exe which is GUI-only due to Windows subsystem limitations).

Interactive Mode

If executed with the `-i` switch, MoneyWorks will enter interactive mode, reading commands from stdin, and writing results to stdout. Any errors will be written to stderr. Diagnostics may also be written to stderr, and will precede the error message. The quit command will end the interactive session, as will (on OS X) closing stdin by sending an EOT (ctrl-D). Any open network connection or document will be cleanly closed first.

Execute Mode

If executed with the `-e` command line switch and a supplied command, the command will be executed and the tool will exit after completion, writing any output to stdout. Multiple `-e` command terms may be passed. They will be executed from left to right. The command in the `-e` term should be quoted to protect it from shell expansion, especially if it contains spaces. A URL or file may be supplied as the final parameter. MoneyWorks will open the file/url before executing the commands. Errors are reported as for the `-i` case.

Output Format

`-h` and `-x` can be used with `-i` or `-e` (if neither of these is specified, `-i` is implied). These output format modes are intended to make the interactive output more reliably machine-parsable.

The `-f` option is useful only on managed Citrix desktops, where you can control the launch environment of the app to restrict users to their own subfolder. It should be used in conjunction with Datacentre server login for server-side enforcement of folder views. It is helpful in that it removes the requirement for the user to manually log in to the server.

OPTIONS

Long form options are only available in the Mac version.

`-i, --interactive`

Run in interactive mode, reading commands from stdin, writing output to stdout and errors to stderr, until command is quit. In verbose mode, success notifications may be appear on stdout for some commands that do not otherwise have output (e.g. “Connect Succeeded”)

`-e cmd, --execute=cmd`

Execute the supplied command and exit.

`-x, --xml`

Output results as xml to stdout (use with `-i` or `-e`).

There is always an xml header line; root tag is `<commandstream>`; each command (including any implicit open) outputs a `<command>` result tag; there is always a `<status>` tag with 'OK' or 'Fail'; If status is OK, there will be a `<result>` tag (may be empty). If status is Fail, there will be an `<error>` tag

and may be a <diagnostic> tag with additional information. Do not rely on the contents of the diagnostic tag. Contents of the result tag are utf-8 and entity-encoded and may be CDATA if it can't be entity-encoded. Stderr may be ignored, although it may contain additional error messages (such as import error logging).

Note: the <commandstream> wrapper is new in version 6, as is the result from the implicit open.

-h, --http

Output results in http-header style to stdout (use with -i or -e).

Each header line consists of the header name (which does not contain spaces), a colon, a space, a value which may contain spaces, and a return+newline. The final header is followed by an additional return+newline (i.e. a blank line). There is always a 'Status' header with value 'OK' or 'Fail'; If Status is Fail there will be an 'Error' header and may be a Diagnostic header (as for xml). If Status is OK and there is output following the headers, there will be a Content-Length header whose value is the number of bytes of content. Stderr may be ignored, although it may contain additional error messages (such as import error logging).

-q, --quiet

Quiet. No initial greeting or success notifications on stdout in -i mode (-x and -h imply -q).

-v, --verbose

Verbose mode. Provide more status feedback on stdout in -i mode.

-o, --readonly

Open document read-only.

-r regonum, --rego=regonum

Use registration number. If you are executing as user www or similar and wish to connect to a Gold server or open a local file, MoneyWorks does not have access to a preferences file to obtain a registration number; in this case you would need to supply one on the command line. Usually, you don't need a registration number to connect to a DC server, although you might like to connect using a full Gold serial to avoid any possibility of all concurrent connections being exhausted.

-u user:pass, --user=username:password

When opening a password-protected local file on command line, use the given username and password.

-a, --activate

Come to front on launch (GUI mode)

-n, --no-manual-connect

Don't allow use of manual connections (Windows GUI mode)

-f folder, --fold=folder

Connect only to named subfolder on Datacentre (Windows)

On Windows, the older, Windows-style switches are also supported (although use of this style of switch is deprecated):

/user=user:pass equivalent to -u

/nomanualconnect equivalent to -n

/fold=foldername equivalent to -f

COMMANDS

The commands you can issue are analogous to the Applescript and COM commands.

Commands are given in the form

verb [param-name "=" DELIM param-value DELIM]*

The parameter delimiter can be pretty much any character you like except for whitespace or =.

On Mac OS X, the command interpreter supports line editing and history.

version

Returns the MoneyWorks Gold version

evaluate expr='expression'

Evaluates the expression and returns the result. You can also abbreviate this to =expression.

open file='URL-or-path' [login="user:pass"]

Open the file or connect to the url, which should follow the moneyworks:// scheme. The MoneyWorks URL scheme is documented here <http://cognito.co.nz/support/faq.php?art=url> If connecting to a URL of a document that requires login, include the login information in the URL, as per the URL scheme. If opening a local document that requires login, then use the option login parameter to supply user name and password.

close

Close the current file or connection, if any

quit

Exits interactive mode

import [data='...' | file='path'] map='path' [additional arguments]

Import data. Since there is no plugins folder in command line mode, the full path to the import map must be given. Likewise if importing from a file, the full path to the file must be given.

There is no metacharacter expansion in the data parameter, so you can't use \t, \n etc.. If you have multiple lines to import, write your data to a temp file and use the file parameter.

Output is a report of number of records created and updated (except for User, Contact and Build files, which will always report zero).

Additional arguments may be:

filename='Account | User | Build | Memo' Import into fixed format file (no import map)

update='true' seqnum='num' [discard='true'] Transactions only. Recall the identified invoice. Optionally discard imported invoice (i.e. just delete or cancel invoice being recalled)

return_seq='true' Output will be sequence number of newly imported record

post='true' Transactions only. Post the imported transactions

post seqnum='sequence number'

Post the transaction identified by sequence number

export table='...' [format='...'] [search='...'] [output='path'] [user_var='value']*

Export data from the given file ("table") in MoneyWorks. If there is no search parameter, you will get all records, otherwise only those matching the search expression.

Using "=" as the search expression will export a single "record" containing just the field names for the file. The table name may include a sort specifier (e.g. "Name.Code-"). To export with a particular format, use the format parameter. Everything in the format string is returned verbatim except for anything inside [...] which is treated as an expression which can reference the fields of the file being exported. Thus if you want tab-delimited, then put tabs between the expressions. You can use metacharacters \t \r \n \xHH (hex) or \\. e.g. "[Code],[Phone]\n" If the output parameter is not supplied, the data is sent to stdout, otherwise a file is created using the full path supplied and the data is written there. If you supply an output parameter and no format parameter, you will get the format equivalent to a manual export, otherwise you get all fields. Using a format parameter is strongly recommended, since default export formats are subject to change.

doreport report='path' format='text|html|pdf' [from='date' to='date'] [output='path'] [user_var='value']* [font='name' size='num' leading='num'] [title='titletext']

Executes the report and writes output to stdout or to the given output file. Since there is no plugins folder in command line mode, the full path to the report must be given. If outputting to stdout, the format is tab-delimited. If a path is given and the filename ends with ".html" then the output will be in

HTML. Otherwise you can explicitly specify the format.

All parameters are inserted into the name table and are available to the report. Thus you can pass values for custom controls defined for a report.

doform form='path' [format='pdf'] [search='expr'] [output='path'] [user_var='value']*

Renders the form to a pdf file. For form types other than .rept a search expression must be provided to select the record(s) to use. If multiple records are selected, all output still goes to the same PDF. Output path is optional — if it is not supplied, a file will be created in the temp directory. The return value on stdout is the path to the generated pdf. The only supported format at present is “pdf”. For forms that use variables supplied by the form printing dialog (such as Message, Print_Copy etc) you may provide values for those variables by passing them as parameters e.g. Message=“Hello” Print_Copy=“0” Include_Remmit=“0” for typical invoice forms; Stmt_Date=“1/1/2010” for statements.

diagnose

Runs file-system and integrity diagnostics for the currently-open file

NOTES

Mac

If executing from a web page on OS X, the MoneyWorks Gold executable must be in a location that is visible to user www and should be executable by everyone (.apps are by default). A symbolic link to the executable will not work, and the executable can't be used in isolation from its .app bundle. This is because access to the resource file is required.

Error reporting

–x and –h modes return a 'status' tag/header indicating success or failure. In the default mode (without –x or –h) errors are written to stderr. Typically one or more errors tagged with “[ERROR]” followed by a failure notice tagged with “[FAIL]”. In –x or –h mode, the final [ERROR] is included in the output to stdout tagged as 'diagnostic', and the failure notice tagged as 'error'.

Performance

Note that connecting to the server is relatively expensive. This will of course be much longer if the server has to open the document from scratch. In command line mode, no plugins are downloaded from the server.

In most circumstances, a short-lived connection to the server with command(s) passed using the –e option will be the easiest mode of use. This is because the calling code can read all of the output to the end of file. In interactive mode you need to know how many bytes of output to read before sending the next command (hence the –x and –h options). If you need a number of results that require some complex processing, consider writing a custom report and executing that using doreport, thus obtaining all the results at once. For highest performance for many requests, you would need to stay connected in –i mode.

If using the –i option, avoid keeping connections open indefinitely, Datacentre should be allowed to close documents overnight for housekeeping purposes. In fact, if you do attempt to stay connected for more than 5 minutes without doing anything, the Datacentre server will treat you as a stale connection and disconnect you. You would need to be pinging the server with some kind of database request, such as an export, to keep the connection alive.

EXAMPLES

(note that these examples are for Mac OS X)

Connect to a DC document and get the output from the given report for the period range.

```
moneyworks -e "doreport report='/Users/rowan//My Balance Sheet' \
from='1/2/10' to='31/3/10' " \
"moneyworks://zaphod.local:6699?doc=Acme.mwd5"
```

Interactive mode (verbose, unformatted output)

```
moneyworks -iv
MoneyWorks Gold 6.0 interactive mode. Type 'quit' or ctrl-D to exit.
open file="/Users/rowan/Documents/Acme Widgets.mwd5"
Open succeeded
=name
Acme Widgets Ltd
close
Close succeeded
quit
```

Piping commands in from a file (such as multiple exports)

```
moneyworks -i < ~/mycommands > ~/output.txt ~/Acme\ Widgets.mwd5
```

BUGS

Launching in GUI mode from the symlink at /usr/bin/moneyworks will actually relaunch using the full path. This appears to be necessary due to limitations in Apple system software. The relaunch via *system(3)* will quote your arguments using double quotes. Arguments that include double quotes will break.

HISTORY

MoneyWorks command line mode first appeared in MoneyWorks Gold 5.0.8

AUTHOR

Cognito Software Ltd <bugreport@cognito.co.nz>

COPYRIGHT

Copyright 1992–2010 Cognito Software Ltd