



## **MacYTDL Video Downloader**

**Version 1.30 – 25 December 2025**

# MacYTDL – MacYTDL video downloader

## Version 1.30 – 25 December 2025

What MacYTDL does .....	5
System requirements .....	6
MacYTDL is free .....	6
How to install MacYTDL .....	7
Using MacYTDL.....	11
More on the Main dialog.....	15
One-time settings .....	15
Batches .....	16
“Show settings before download” .....	16
Using MacYTDL on Apple Silicon Macs .....	17
Settings.....	18
“Change download folder:” .....	18
“File format” .....	19
“Maximum resolution” .....	19
“Download subtitles” .....	20
“Subtitles format:” .....	20
“Subtitles language”.....	20
“Embed subtitles” .....	20
“Auto-generated subtitles” .....	20
“Download description”.....	21
“Audio only” .....	21
“Audio format” .....	21
“Write thumbnails” .....	21
“Embed thumbnails” .....	21
“Add Metadata” .....	21
“Convert” .....	21
“Keep original video and/or subtitles file” .....	22
“Get formats list” .....	23
“Force overwrites” .....	24
“Use proxy”.....	24
“Use cookies” .....	24
“Use custom template” .....	24
“Use custom settings” .....	26
Admin.....	27
“Verbose logging” .....	27
“Add timestamps to log” .....	28
“Check YT-DLP version on startup” .....	28
“Show download settings before starting download” .....	28
“Automatic download” .....	29
“Limit download speed” .....	30
“Download videos in parallel” .....	30
“QT compatible video” .....	31
“Discard URL after download” .....	32
“Don’t use “part” file names in download” .....	32
“Delete partially downloaded files” .....	32

“Use .netrc credentials” .....	32
“Suppress warnings from YT-DLP” .....	33
Utilities.....	34
“Open log folder” .....	34
“Open download folder” .....	34
“Update YT-DLP” .....	34
“Open YT-DLP web page” .....	35
“Switch to YT-DLP” .....	35
“Save current settings” .....	35
“Restore settings” .....	35
“Return to default settings” .....	35
“Check for MacYTDL update” .....	35
“Check for FFmpeg update” .....	36
“Switch to FFmpeg to [Intel/ARM]” .....	36
“[Install Deno/Check for Deno update]” .....	36
“[Install/Remove] Atomic Parsley” .....	36
“[Install/Remove] Service” .....	36
“Delete logs” – command-d .....	36
“Uninstall” – command-shift-U .....	36
“About MacYTDL” – command-a .....	37
Keystroke shortcuts.....	38
Main dialog .....	38
Settings dialog .....	38
Admin dialog.....	38
Utilities dialog.....	38
About MacYTDL dialog .....	39
Settings shown before download .....	39
Credentials dialog .....	39
Batch functions dialog .....	39
Monitor dialog.....	39
The MacYTDL Service – “Send-URL-To-MacYTDL” .....	40
Download log files .....	43
Using the default “config” file .....	44
How to uninstall MacYTDL .....	45
Using MacYTDL on versions of macOS prior to 12 Monterey.....	46
YouTube URLs and playlists/channels .....	47
Single video.....	47
A playlist .....	47
Single playlist item.....	47
A playlist item .....	47
A channel.....	48
Live stream.....	48
An uploader playlist .....	48
A YouTube mix .....	49
Personal “Watch Later” playlist .....	49
Downloading videos from ABC iView (in Australia) .....	50
Automatic download – Choosing iView episodes .....	51
Working with dual monitors.....	52
Batch functions .....	53

Overwriting existing videos ? .....	56
“Overwrite” .....	56
“New name” .....	56
“Return” .....	56
Using Homebrew and MacPorts with MacYTDL.....	57
“Shortcuts” .....	59
What if the download doesn’t work ? Troubleshooting .....	61
What if internet connection is lost during a download ? .....	61
Limitations and known bugs .....	62
Future developments .....	63
Why use a video downloader ? .....	64
MacYTDL components.....	66
DialogToolkitMacYTDL.scptd .....	66
Myriad Tables Lib.scptd.....	66
Main.scpt .....	66
Monitor.scptd .....	66
adviser.scptd .....	66
batch.scptd.....	66
Utilities.scptd .....	66
Formats.scptd .....	66
Auto_download.scpt .....	66
Send-URL-To-MacYTDL.workflow .....	67
Localisation files .....	67
External components.....	68
YT-DLP .....	68
Deno .....	68
Python scripting language .....	68
FFmpeg .....	68
FFprobe .....	69
AtomicParsley .....	69
Alerter .....	69
ets .....	69
Any comments, suggestions or better code ? .....	70
Software Licenses .....	71
Acknowledgements .....	72
Code signing and notarisation .....	73

## What MacYTDL does

MacYTDL is a utility which downloads videos using the [YT-DLP](#) video downloader. *YT-DLP* is able to download videos from a great many web sites. MacYTDL can be used on Macs running macOS 10.10 Yosemite and higher.

Features:

- Download single and multiple videos, playlists, YouTube channels, live streams and batches.
- Works with YouTube and many other sites. This is a full list: [YT-DLP supported sites](#).
- Variety of download file formats – maximum video resolution can also be set.
- Optionally download and/or embed subtitles in chosen format (from seven) and language.
- Optionally download a text description of the video.
- Optionally download or extract separate audio file in a range of formats.
- Can download multiple videos at one time i.e. in parallel in separate processes, or in succession in one process i.e. in series.
- Automatically resumes incomplete downloads and has a pause/resume function.
- Options for login credentials, proxy, cookies file, custom settings, custom file name template (thanks to [tbt7](#)), Quicktime compatibility, suppress warnings, use “.part” files, discard downloaded URLs, delete partially downloaded files and to limit download speed.
- Can show a list of files available for download (called formats) – compatible formats can be merged.
- Is localised – currently available in Italian (thanks to “[bovirus](#)”), Spanish (thanks to [michel-GH](#)), French (with help from Didier), German and Portuguese (thanks to Google Translate). Some translations are approximate. Some new text is shown in English.
- Options to embed thumbnail images and metadata in downloaded files.
- Optional [macOS Service](#) can send a URL to MacYTDL or download from the URL using current settings (Automatic download).
- Facility for saving, restoring and returning settings to default.
- When a download is complete, MacYTDL offers to play file in default player.
- External components downloaded and installed by MacYTDL, controlled by the user.
  - Can check for updates to MacYTDL and components.
  - *YT-DLP* can be checked for updates automatically on startup.
- A separate log file is saved for each download enabling problem solving.
- Has a built-in uninstaller which moves all MacYTDL related files to the Trash/Bin.
- Runs natively on Intel and ARM Macs – i.e. Rosetta 2 is not required on M1/M2/M3 Macs.

## System requirements

MacYTDL will work on any Apple Mac running OS X 10.10 Yosemite or later. However, there are limitations for users on macOS prior to 10.15 Catalina (see [more detail](#)).

## MacYTDL is free

MacYTDL is a retirement project and will always be free.

Please consider sponsoring the *YT-DLP* team. Detail on sponsorships is available here:

[pukkandan](#), the owner of the *YT-DLP* repository although not currently a lead maintainer. Some of the tool's [maintainers](#) and [contributors](#) can also be sponsored.

Similarly, MacYTDL would not be possible without *FFmpeg*. Detail on making donations to the *FFmpeg* team are available here:

<https://ffmpeg.org/donations.html>

Other developers who should be supported include:

[Martin Riedl](#), who provides macOS binaries of *FFmpeg*.

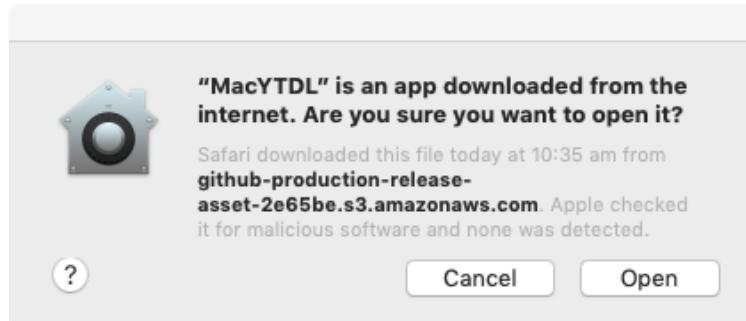
[Helmut K. C. Tessarek](#), who also provides *FFmpeg* binaries.

[Valere Jeantet](#), author of the *Alerter* tool.

## How to install MacYTDL

Download and open the DMG file. Drag the MacYTDL applet to *Applications* folder. That will make it available for all user accounts. MacYTDL does not work properly if run from within the DMG file. It's best to delete old copies of MacYTDL – replace with a new copy or move old copies to the Trash/Bin.

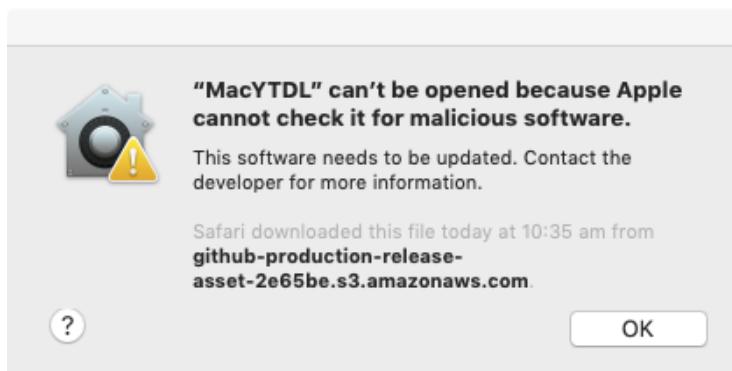
The applet is code signed and notarised so, you may see messages when you first run MacYTDL. The range of messages varies with the version of macOS being used. In macOS Mojave and Catalina this message is shown when MacYTDL is first run:



Click on “Open”.

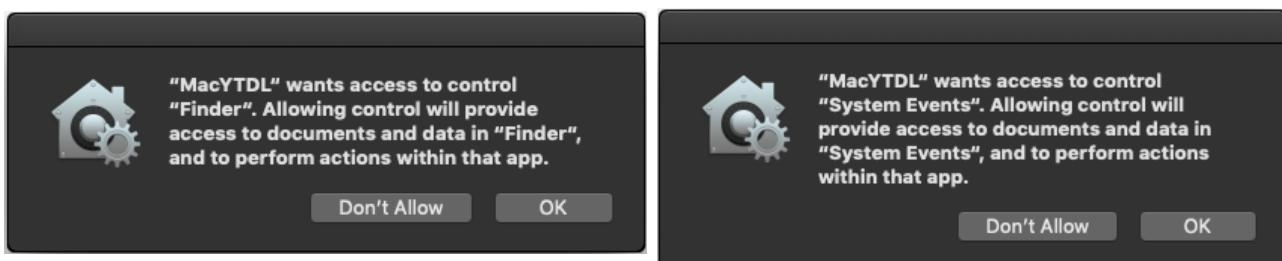
Detail on code signing and notarisation is [available below](#).

If you see this message instead ...

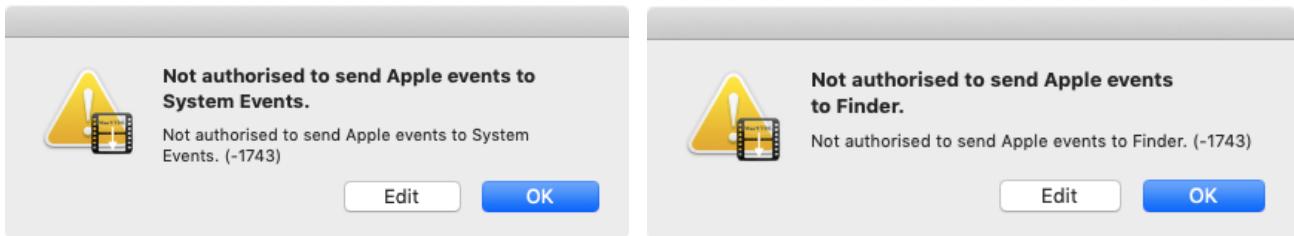


... you will need to change your Gatekeeper settings to allow applications from “identified developers”. For information on Gatekeeper and how to change settings visit the Apple web site at: <https://support.apple.com/en-au/HT202491>.

Again depending on which version of macOS you are using, there may be a number of messages requesting permission for MacYTDL to control the Finder and System Events:



Alternatively, you may see these messages:

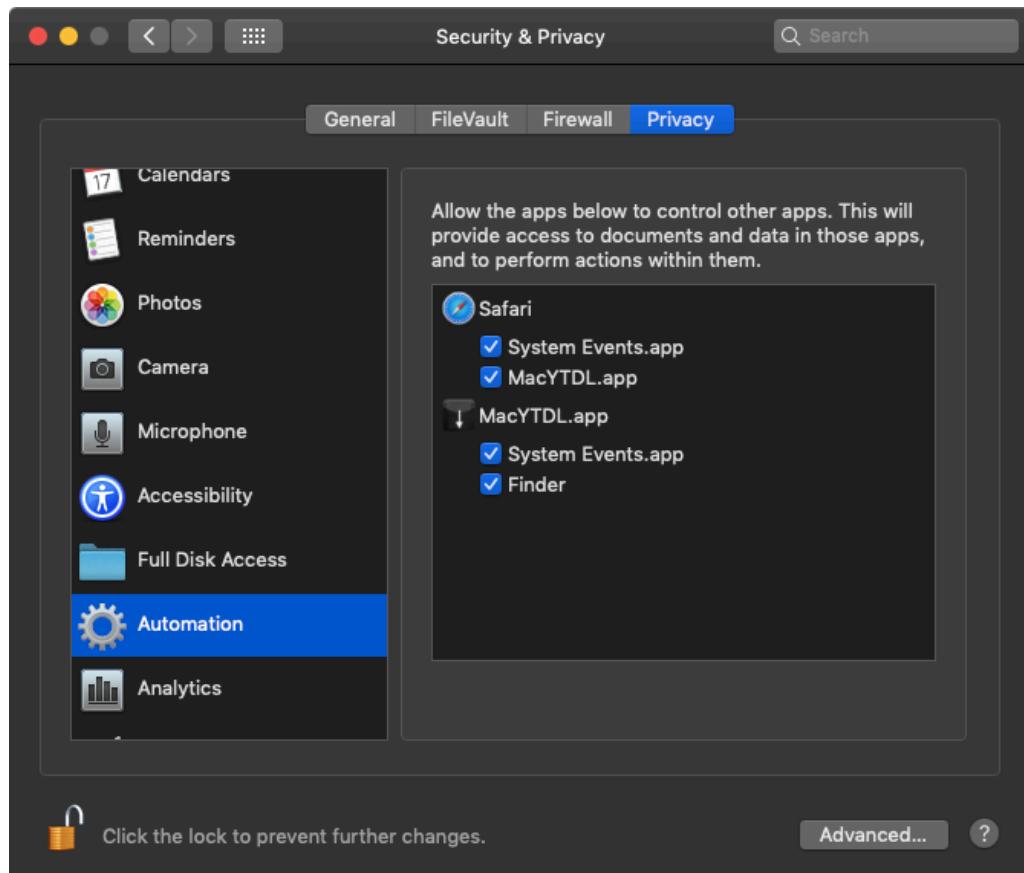


MacYTDL cannot function without those permissions, Click on “OK”.

If you see the second kind of message or you click on “Don’t Allow” on the first, you will need to give permission manually in System Preferences:

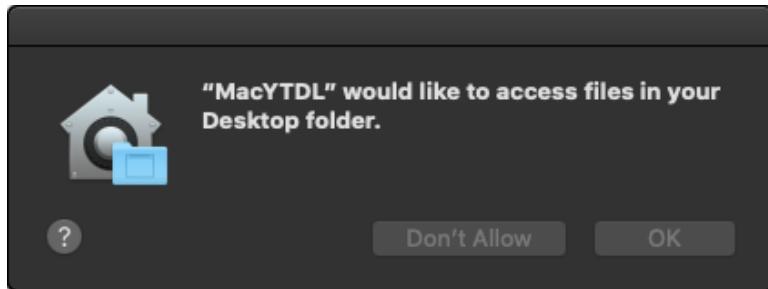
- go to *System Preferences* on your Mac;
- choose the “Security & Privacy” preference pane;
- choose the “Privacy” tab; and
- click on the “Automation” entry in the left hand sidebar.
- If the two boxes under “MacYTDL” are not ticked:
- click on the lock symbol;
- enter an administrator user name and password; and
- place a tick in each box.

Your permissions should look like these:



In the above example, there are two tick boxes under “Safari”. They are required to use the [MacYTDL Service](#).

In macOS 10.14 and higher, the permission dialogs may be shown each time you open a new version of MacYTDL for the first time. While using MacYTDL, other dialogs requesting permission may be displayed. For example:



Display of permission dialogs depends on the location of the downloads folder, which MacYTDL components are installed and which version of macOS is being used.

Always click on “OK”. Otherwise, MacYTDL may not be able to access files such as the logs and downloaded videos.

The first time MacYTDL is started you will also be advised that various components used by MacYTDL need to be installed:

- A new folder called “MacYTDL” will be created in your “~/Library/Preferences” folder. It will contain a preferences file for MacYTDL, a “simulate” file used to check URLs and get file names, log files produced by *YT-DLP* and any batch file.
- *YT-DLP* will be downloaded and saved in the folder “/usr/local/bin/”. The folder “/usr/local/bin/” will be created if it doesn’t exist. For users on macOS 10.15+, another folder, “/usr/local/bin/\_internal”, will be created to hold *YT-DLP* support files.
- Three utilities, *FFmpeg* and *FFprobe*, will be downloaded and saved also in “/usr/local/bin/”.
- MacYTDL will offer to install the *Deno* Javascript runtime which is recommended by *YT-DLP* to download from YouTube. *Deno* is not required to download from other sites. It is stored in the “/usr/local/bin” folder. If *Deno* is not installed, MacYTDL will offer to install it on the first occasion a download from YouTube is started. If *Deno* remains not installed, an installer will be available in the Utilities dialog.
- A script library which provides code for MacYTDL’s dialogs will be copied to “~/Library/Script Libraries”. That folder will be created if it doesn’t exist. The script library will be automatically updated as new versions are released.
- MacYTDL will offer to install a Service in your “~/Library/Services” folder. The Service called “Send-URL-To-MacYTDL.workflow” sends URLs to MacYTDL from other applications. It is not required for MacYTDL to function and can be installed later if preferred. If installed, the Service will be automatically updated as new versions are released.
  - [More detail is provided below](#).

**Please note:** downloading the various components takes time. It can look as if MacYTDL is not doing anything. Also, the component downloads cannot be cancelled. **Please just wait.**

Zip files will be saved into “/usr/local/bin/” during the process to install *FFmpeg* and *FFprobe*. **Do not delete them.** That will cause an error. The two Zip files will be automatically deleted when the install is finished.

On every startup, MacYTDL will automatically look for and re-install any missing components. When installation is finished, the MacYTDL Main dialog will be displayed.

## Using MacYTDL

MacYTDL will try to download a video from any valid URL that is entered into the “Paste URL here” text box. URLs can be:

- copied from any app and pasted into the text box;
- clicked-and-dragged from any app into the text box;
- typed manually into the text box;
- pasted into the text box by the [MacYTDL Service](#).

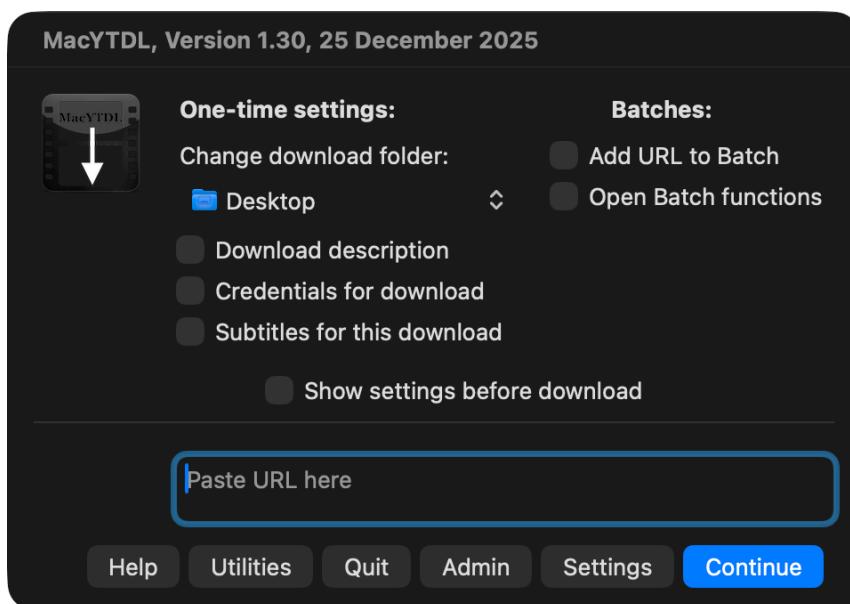
When a URL has been entered into the Main dialog, click on “Continue” or press the [Return] key to start the download.

A log file is saved for each download which contains a report from *YT-DLP* on the various steps in the download process. The log files can help particularly in diagnosing problems. They are named:

ytld\_log-[name of download file]-[date-time download initiated].txt

Log files can be opened in any text editor or word processor.

This is the Main dialog:



If you have installed the MacYTDL Service, instead of copying the URL you can choose:

[application menu eg. “Safari”] => Services => Send-URL-To-MacYTDL

A keystroke shortcut can be assigned to the MacYTDL Service using the macOS Keyboard Settings.

The MacYTDL Service will check whether you have selected a URL in the current document, have copied a URL to the Clipboard or have opened a web page in a web browser, switch to MacYTDL and paste the URL into the text box.

Multiple URLs can be pasted or entered into the text box before downloading. They can be separated by space, comma, return, tab or linefeed characters. MacYTDL will download each video in turn in

one process (and saving just one *YT-DLP* log file). Alternatively, successive URLs can be pasted and downloaded serially. Also multiple URLs can be pasted and videos downloaded in separate parallel processes. Current settings can be displayed before starting a download.

After initiating the download, the Main dialog will disappear for a while. MacYTDL will do a test download to check the URL and if it's OK, will redisplay the Main dialog, start the download and display a progress "Monitor" dialog. Sometimes, the Monitor dialog is displayed before any downloaded data has been saved. In that case, the Monitor dialog reports "Progressing" but as the download continues, it will report on progress:



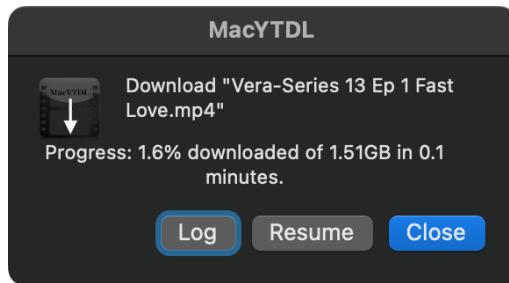
*Note:* It can take a long time for MacYTDL to check a URL if it is complicated or is for a playlist.

From the Monitor dialog, you can open the log folder, stop the download, pause the download or close the dialog.

To cancel a download, click on the "Stop" button. The download will cease and, optionally, any partially downloaded files moved to the Trash/Bin. This will also cancel downloading multiple videos that are in one process (e.g. YouTube playlists or parallel downloads). Completed downloads, subtitles files, description files and partially completed batch downloads are not deleted.

To pause a download, click on the "Pause" button. The download will stop but any partially complete files will remain in place. The "Pause" button will be replaced by the "Resume" button.

To resume the download, click on the "Resume" button:



It can take some time for the download to resume. When the download resumes, a new Monitor dialog will be displayed. Also, a new log file will be saved recording progress with the resumed download. If the "Close" button is clicked after download is paused, the download will remain incomplete. If the partially complete file is retained in place, the download can be resumed at a later time by initiating the download in the Min dialog as before. Partially downloaded files are can be opened in video players.

To download a paused video at a later time, click on "Close" and when desired, send the relevant URL to MacYTDL again. The download will resume from the point at which it was paused.

*Note:* MacYTDL cannot stop, pause or resume a download after the Monitor dialog has been closed. Use the macOS *Activity* Monitor if you need to cancel a download after closing the Monitor.

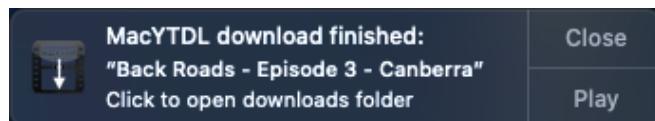
You can leave the Monitor dialog open and switch back to the Main dialog to download another video. A separate Monitor dialog will be shown for each download process. Each can be closed and the related downloads stopped separately in any order.

Monitor dialogs provide some detail on progress with the download. Depending on the site from which a video is being downloaded, progress is shown as a percentage of the total expected file size or the total downloaded up to that point. The time elapsed since download started is also shown. Some downloads involve a number of separate files e.g. playlists and separate video-audio merges. In those cases, the progress reported will reset when the next file download commences. Monitor dialogs are updated every 5 seconds.

Monitor dialogs cannot be hidden or minimised.

Downloaded files are named as usefully as possible with series name and episode details for the Australian broadcasters ABC (iView), TEN (10Play), SBS (On Demand), Nine (9Now), Seven (7Plus) and for ITV. Other download files are given the name provided by the uploader.

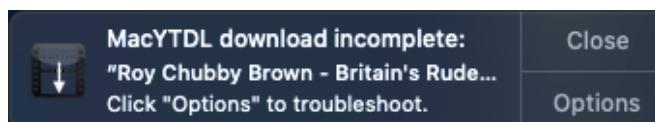
MacYTDL issues a notification when each download process is finished:



Clicking on the body of the notice will open the downloads folder. The "Play" button will open the downloaded video in the default video player for the format of the file downloaded. If the notification is for a playlist or other multiple download, the first video in the list will be played. The notification will automatically quit after 10 seconds.

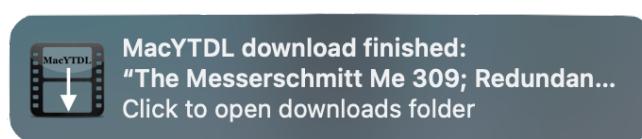
The notification also advises if any errors were reported during the download whether or not the was download completed.

There are two options: to open the log file of the download concerned; and to open the MacYTDL log folder.

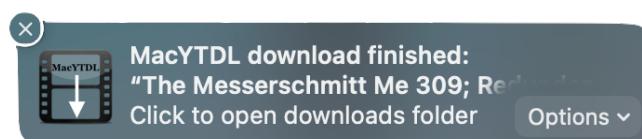


The "download incomplete" notification will automatically quit after 20 seconds.

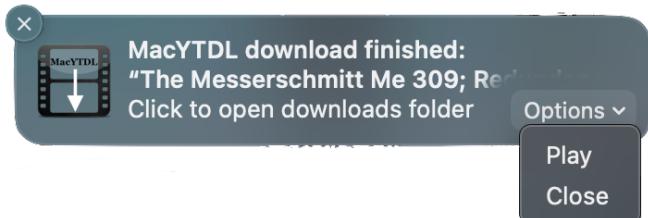
In macOS 11 Big Sur and higher, notifications will look different but behave in much the same way. This is a notification of completed download:



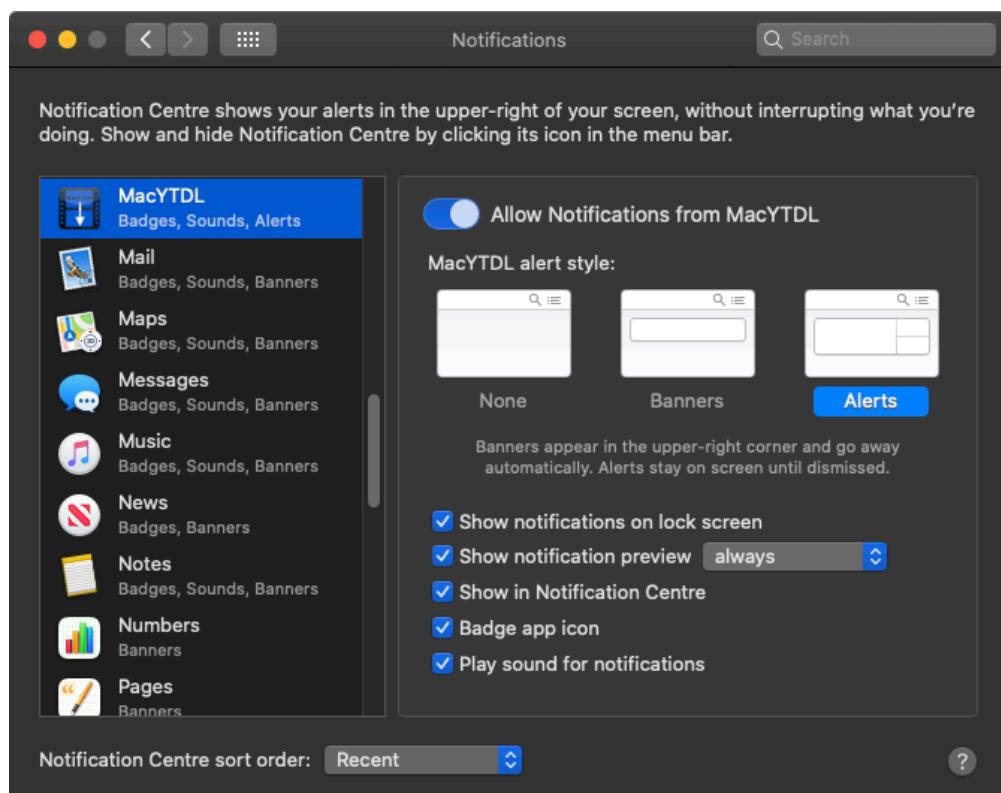
Hovering your mouse over the right hand end of the notification to see the available options:



Click on the "Options" button to access additional functions:



The form of notifications can be controlled in the “System Preferences”. As with all macOS notifications, there is a choice of alerts, banners or no alert and MacYTDL notifications can be disallowed. MacYTDL notifications, however, do not play a sound.



## More on the Main dialog

The Main dialog should be reasonably self-explanatory. Paste or enter a URL into the text box and click on “Continue” or press the Return key to start a download.

The position of the Main dialog will be remembered if it is moved to another place on the Desktop.

Multiple URLs can be pasted into the box, separated by space, return, tab or linefeed characters. This makes it easy to copy a number of URLs from source documents and paste them directly into the MacYTDL text box. When “Continue” is clicked, those videos will be downloaded sequentially in one process.

If preferred, multiple files, batches and playlists, can be downloaded in parallel, separate processes. That can be specified in [the “Admin” dialog](#).

*Note:* With multiple downloads and playlists, if an error is encountered with a download, *YT-DLP* will continue with the next download file.

Click on “Settings”, or press command-s, to change the MacYTDL settings. See [below for details](#).

Click on “Admin”, or press command-a, to change how MacYTDL functions. [See below for details](#).

Click on “Utilities”, or press command-u, to access a variety of functions. See [below for details](#).

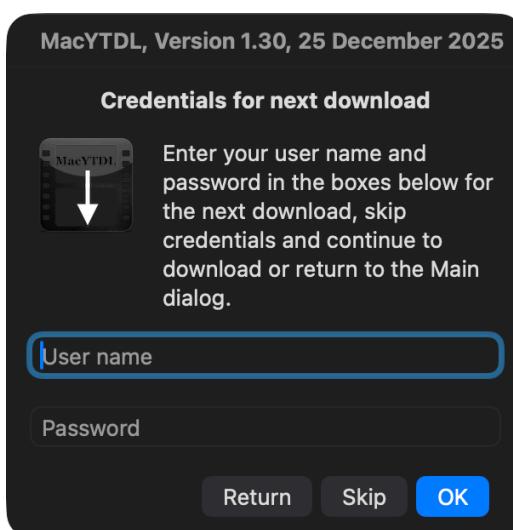
To show this help file click on “Help”, or press command-shift-?

Click on “Quit” or press command-q to quit MacYTDL.

### One-time settings

Five options can be set for the next download. [These settings are not saved](#).

- The download folder can be set.
- The download container can be remuxed, that is, converted into another format. The choice of format is: mp4, mkv, webm, ogg, avi and flv. Remuxing is always done by *FFmpeg*. This can be handy when the desired container format is not available from the web site. The original file can be kept or automatically deleted according to the choice in Settings.
- A file containing a description of the video (if available) can be downloaded and saved with the video file. The file will be named “[the video name].txt”.
- Username and password credentials can be added for the next download:



- A subtitles file can be downloaded (the subtitles format and language are taken from “Settings” or set to the default, “best”).

The One-time settings return to the saved settings after the “Continue” button is clicked. Thus, the user may have set the usual download folder to “~/Movies/” but can use the runtime setting to download the next video to somewhere else eg. “~/Desktop/”. After that videos will be downloaded to “~/Movies/”.

### **Batches**

Instead of downloading immediately, URLs can be saved for downloading later. When the “Add URL to Batch file” checkbox is ticked and the “Continue” button is clicked, the URL is saved in a “batch” file. The batch file is stored in the MacYTDL preferences folder.

To access all batch functions, click on the “Open Batch functions” checkbox and click on “Continue”. See [below for more detail](#).

### **“Show settings before download”**

If ticked, current download settings will be displayed with the options to quit, edit settings, cancel the download or download. The dialog also provides the option to change the saved “Show settings before download” setting.



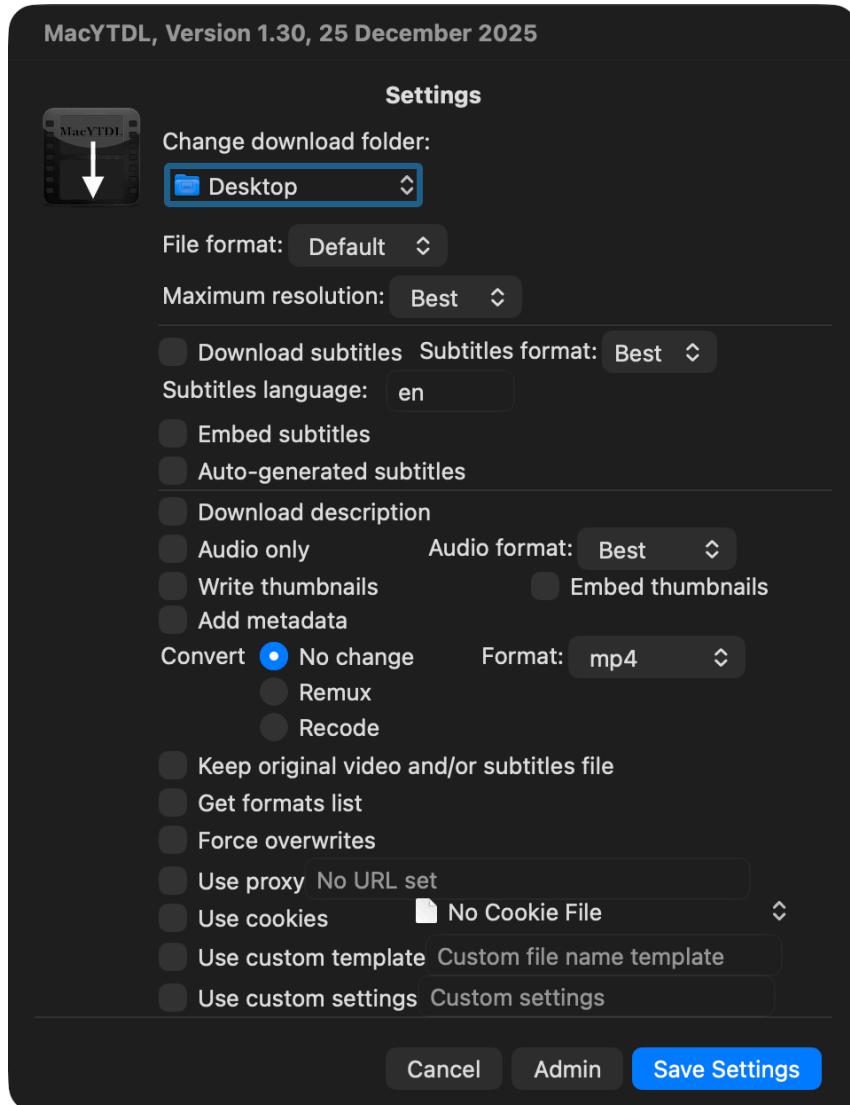
## Using MacYTDL on Apple Silicon Macs

Apple's move from Intel to Apple Silicon CPUs (i.e. M1/M2) affects MacYTDL. The *FFmpeg* utility is used to provide downloading and conversion functions. To avoid using Rosetta 2, MacYTDL checks the architecture of the Mac on which it is running. It will install the ARM64 version of *FFmpeg* and *FFprobe* on M1 and later Macs. It will install the x86-64 version of *FFmpeg* and *FFprobe* on Intel Macs.

There is a facility in [Utilities](#) to switch between the Intel and ARM versions of FFmpeg for users on Apple Silicon Macs.

## Settings

MacYTDL uses a number of options that are provided by *YT-DLP* to control downloads. To access settings, click on the “Settings” button in the Main dialog or press command-s. The Settings dialog will be displayed:



### ***“Cancel” – command-***

Exit and return to the Main dialog without saving any changes. Keyboard shortcut is command-d.

### ***“Admin” – command a***

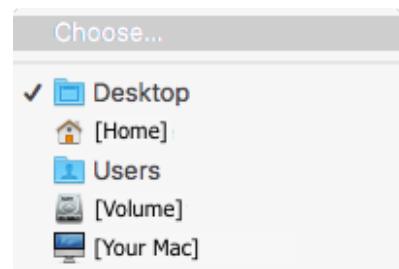
Saves changes to options and shows the “Admin” dialog.

### ***“Save Settings” – Return***

Save all changes to options and return to the Main dialog. Keyboard shortcut is Return.

### ***“Change download folder:”***

To change the folder in which videos are saved, click on the drop-down box and click on “Choose...”. Choosing any of the other



locations listed will not change the download folder.

A choose folder dialog is shown enabling you to select a folder in which downloaded videos will be saved. To create a new folder, press command-shift-n or (macOS Catalina and later) click on the new folder icon in the dialog's toolbar.

*Note:* The download folder must be available for MacYTDL to be able to save videos into that folder. If the download folder is on an external volume, make sure that volume is mounted on your Desktop. Also, MacYTDL might not be able to save videos on an external volume which is password protected.

### ***“File format”***

By default, MacYTDL will try to source files from the given web site which can be merged to give the highest resolution video and best quality sound together. If the best quality available is a single file, that will be downloaded. This setting forces MacYTDL to download the best single file available in the chosen format. It can result in a lower quality, often smaller, file which often is not desirable. Thus, this setting can usually be left at “Default”.

*File format* specifies a choice of container format. The choices are: Default, mp4, webm, ogg, 3gp and flv. MacYTDL checks whether the format chosen is available from the web site. If not, it asks whether to:

✓ Default
mp4
webm
ogg
3gp
flv

- ignore the preference and download the best format available;
- download the best format and remux to the preferred format;

*Note:* Remuxing to some formats will fail if the downloaded video stream is not encoded with the required codec. Remuxing to “3gp” is not available and will revert to “mp4” instead;

- cancel the download.

*Note:* This setting is effectively over-ridden by the “Convert” setting. So, for example, if “File format” is set to “mp4” and “Convert format” is set to “mkv”, MacYTDL will download the best mp4 video file and merge that with the best audio file into an mkv container. By default it will then delete the separate video and audio files.

### ***“Maximum resolution”***

✓ Best
2560
1920
1080
720
480

This setting sets the highest resolution to be considered for download. By default, *YT-DLP* looks to download the best video and audio files available. Higher resolutions (e.g. 1080p) are preferred to lower resolutions (e.g. 720p). If “Best” is selected, *YT-DLP* will make that default choice. The measure used is the vertical resolution.

If “Maximum resolution” is set to a lower value, e.g. “720”, files with that resolution or the next lower resolution will be downloaded. If that and smaller resolutions are not available, the next highest resolution available will be downloaded.

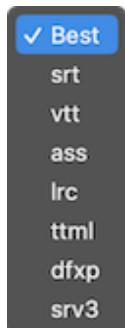
This setting can be useful when videos are to be viewed on a low resolution screen or on a slower, old Mac which might have difficulty with 1080p or higher files.

## ***“Download subtitles”***

Enables subtitles file to be downloaded if they exist. Subtitles are saved in the chosen format and converted if the web site does not provide that format. If subtitles are not available, MacYTDL provides choice to cancel or continue the download without subtitles. If the video is being downloaded from YouTube, this setting will provide author-generated subtitles if they are available. MacYTDL will warn if author-generated subtitles are requested but only auto-generated are available and vice versa.

## ***“Subtitles format:”***

Enables a choice of subtitles format. Choices are: best, srt, vtt, ass, lrc, ttml, dfxp and srv3. If the chosen format is srt, ass, vtt or lrc but is not available, the best available will be automatically downloaded and converted to the desired format. If the chosen format is dfxp, srv3 or ttml and that format is not available, MacYTDL will download the best available.



## ***“Subtitles language”***

Enables choice of subtitles language. To request a language, enter the relevant language code. A list of [language codes is available on Wikipedia](#). Use the two-letter codes in the “639-1” column. Note, however, that some web sites adhere to the IETF standards which can include country or region codes e.g. “en-US” for English US. If a web site does not offer the language you request, MacYTDL presents a list of available languages.

## ***“Embed subtitles”***

Subtitles will be embedded in the video file instead of being saved in a separate file. This function defaults to “author” generated subtitles. To embed “auto” generated subtitles, turn on the “Auto-generated subtitles” setting.

*Note:* Embedding may fail if specified at the same time as a Remux although remuxing to “mkv” format is usually successful.

*Note:* By default, with YouTube videos, MacYTDL will only embed author generated subtitles i.e. it will ignore auto-generated subtitles. To embed auto-generated subtitles, turn on “Auto-generated subtitles”.

*Note:* The subtitles file will be downloaded, then deleted or retained after embedding is finished according to the “Keep original....” setting (see below). Embedding may remove custom formatting eg. text colour.

## ***“Auto-generated subtitles”***

The YouTube web site offers automatically generated subtitles (called “captions”) for some videos. This setting will download those subtitles if they exist.

If both author and auto-generated subtitles are requested and both exist, author-generated subtitles will be downloaded (as they usually have a better translation into text). MacYTDL will warn if auto-generated subtitles are requested but only author-generated are available and vice versa.

### ***“Download description”***

Enables descriptive text provided by the video web site to be downloaded. The description is saved in a text file called “[the video name].txt”. It is saved in the downloads folder.

### ***“Audio only”***

Requests a download of just the audio part of the video. Some web sites offer separate audio files for download. If a separate audio file is available, it will be downloaded in the best quality available. If a separate file is not available, the full video will be downloaded and the audio component extracted to a separate file with the best quality and the original file will then be deleted. If a remux format is set, the downloaded container will be remuxed to the desired format if it is compatible.

*Note:* Remuxing an audio-only download can be tricky and any incompatible combination will cause an error. Usually, there is no need to remux audio files as they are already in a compatible container.

### ***“Audio format”***

If “Audio only” is chosen, the resulting download will be in this format. Choices are Best, aac, alac, flac, mp3, m4a, opus, vorbis, wav.

*Note:* If a remux setting is also chosen, the resulting audio stream will be the most compatible with the selected container format. Thus, for example, if the container is remuxed to “mp4”, the audio will be stored in “aac” format even though mp3 was selected.

### ***“Write thumbnails”***

If it exists, a separate thumbnail image file will be saved in the downloads folder with the video file.



### ***“Embed thumbnails”***

If it exists, a thumbnail image will be embedded in downloaded files. It’s useful especially for music tracks (e.g. album cover art). For users on macOS 10.14.6 and earlier who are using *youtube-dl*, the option is only available if the *AtomicParsley* tool has been installed.

*Note:* Thumbnail images embedded in audio files are not usually visible in Finder. However, they are shown in tools such as *VLC*.

### ***“Add Metadata”***

Any available metadata will be embedded in the downloaded file(s) in the correct variables.

### ***“Convert”***

Audio-video files are, logically, usually a combination of a video component and an audio component (often called “streams”) within a container. Streams are encoded with “codecs”. Common video codecs include: H.264, vp09, H.265 and DivX. Common audio codecs include: aac, mp3, FLAC, Opus and WAV. Common container formats include: mp4, mkv, flv, webm, mov, avi and ogg. Some combinations of streams and containers are more common than others e.g. audio within mp4 containers is usually in aac format. Also, some audio formats can exist without a container e.g. mp3. Some other audio formats are stored within a simple container with the same name e.g. FLAC. Further, the file extension can vary with the kind of device on which the file is played e.g. mp4 audio with Apple’s DRM protection are named “.m4p”.

In MacYTDL, converting enables video downloads to be either remuxed or recoded. Remuxing is the process of converting the video container format. Recoding is the process of re-encoding the streams inside a container as well as converting the container. An example remux would be to convert a YouTube webm file to mp4 leaving the streams unchanged. An example recode would be to convert the video stream from AV1 to H264 and the audio stream from opus to aac before converting the container from webm to mp4. Remuxing and recoding are handled by *FFmpeg*. When recoding, *FFmpeg* will choose the most compatible codecs for the container format specified.

MacYTDL implements remuxes with no change to the audio and video codecs, which are copied to the new container. Thus, it is possible that the audio and video components might not be compatible with the container format. Detail on containers and codecs are available [on Wikipedia](#).

On occasion, the desired container format is not available from a web site. For example, most YouTube videos are in “webm” format which might not suit all situations. With the Remux option, MacYTDL will download the best quality available then convert the download into the chosen container format. The dialog lists video container formats (mp4, mkv, flv, webm, mov, avi and ogg) then audio container formats (aiff, flac, m4a, mka, mp3, ogg, wav). Remuxing is skipped if the downloaded file is already in the chosen format.

There can be situations in which the codecs within the container are not suitable. The “Recode” function will convert the video and audio streams codecs into the codecs which most suit the chosen container.

*Note:* Often it’s best to leave the file format and Convert settings to “No change”. That will ensure *YT-DLP* downloads the best quality video and audio in the most compatible container.

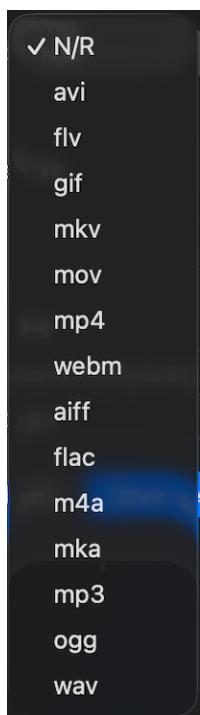
*Note:* MacYTDL does not check for compatibility of available codecs and chosen containers. Details on errors are provided in MacYTDL log files.

*Note:* Remuxing to ogg or webm formats will often fail. Webm is a container for VP8/VP9/AV1 (video) and Vorbis/Opus (audio) formats and will fail in MacYTDL if a stream in one of those codecs is not in the dowloaded file. Similarly, ogg is a container for Speex, Vorbis, Opus, FLAC, Theora, Daala, Tarkin and Dirac formats and will fail in MacYTDL if a stream in one those codecs is not served. If there is an error, the MacYTDL log file should have information from *FFmpeg* on the reason.

*Note:* Remuxing is not available for users of *youtube-dl*. Instead, when this settings is on, MacYTDL will perform a “recode” which can take considerably longer. Also, the “mov” and audio formats are not available.

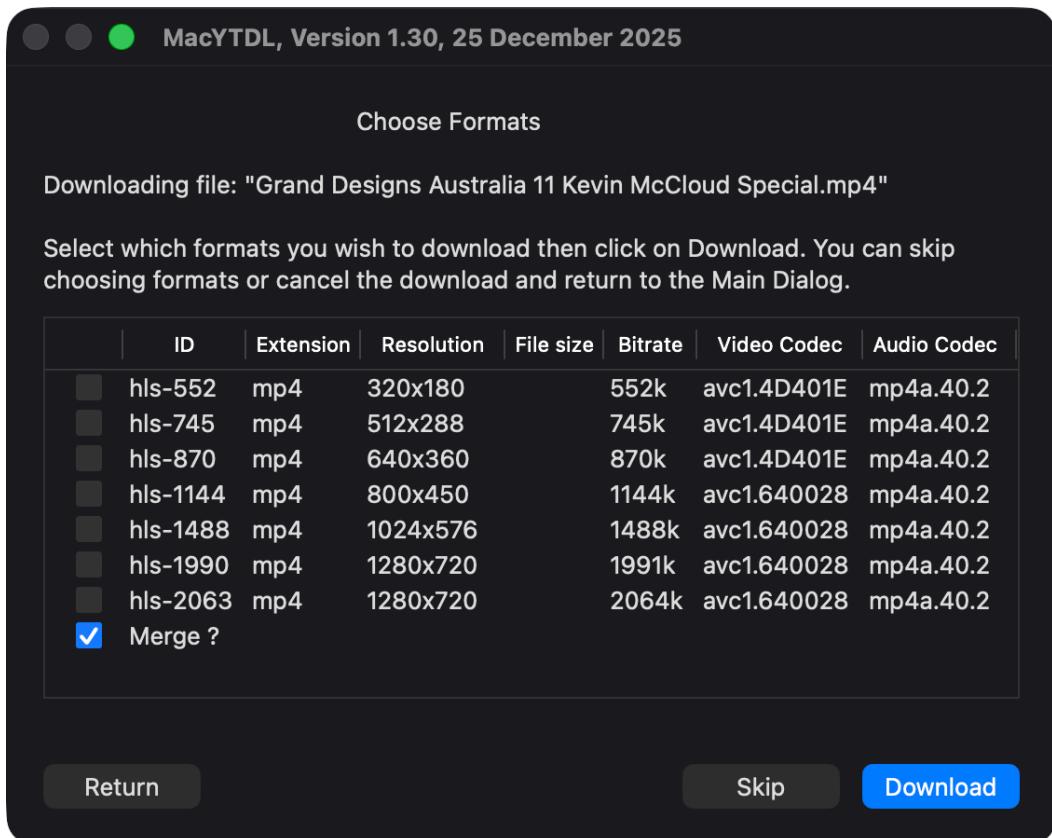
### ***“Keep original video and/or subtitles file”***

*YT-DLP* often perform conversions after downloads have finished, called “post-processing”. For example, to achieve the best quality separate audio and video files might be downloaded then merged. Similarly, MacYTDL has options to remux downloads and embed thumbnails and subtitles etc. This setting enables all the original file(s) to be retained instead of automatically deleted. This can be useful if a custom conversion is required after the download.



### **“Get formats list”**

Many web sites provide a range of downloads for each video. For example, some sites provide files of varying resolution to suit users’ download speed. So, a low resolution file might be served to a user on a slow broadband connection. The “Get formats list” function presents a list of those files so that the user can choose exactly which file(s) to download rather than relying on the default chosen by MacYTDL. In *YT-DLP* documentation, the files listed are called “Formats”:



Web sites often provide separate formats for audio and video streams. With “Get formats list” the formats can be downloaded separately. When the download is complete, the formats can be retained separately or merged into a suitable container. If a particular container format is desired, the format can be set using the “Remux format” MacYTDL setting.

To use the “Get formats list” function, turn on the relevant setting in “Settings” then click on “Save Settings”. Enter a URL in the Main dialog and click on “Continue” as usual. After a pause, MacYTDL will show the “Format chooser” dialog.

*Note:* The “Format chooser” can be long, especially for YouTube videos and so it is scrollable.

To download the desired files tick the box beside the relevant formats then as desired tick the box beside “Merge formats ?”. Usually, it will be best to merge the formats so that they are a playable file containing video and audio. The container format is chosen automatically. However, that automatic choice can be overridden by the “Remux format” setting. It might be necessary to scroll the dialog to the bottom to show the “Merge formats” tick box.

*Note:* The “Get formats list” can also be used to choose the desired format of a live stream.

*Note:* If the container format set with “Remux format” is not compatible with the formats selected, *YT-DLP* will report an error in the log file. Usually the automatically set container format is best.

*Note:* The “Get formats list” can be used to download two or more video or audio formats but not if “Merge formats” is selected. If two video formats are selected to be merged and only the first choice will be downloaded. Similarly, only the first chosen audio format will be downloaded if two audio formats are selected for merging.

*Note:* “Get formats list” setting is ignored by automatic downloads.

*Note:* If three or more formats are chosen for merger, the first two will be merged and the others ignored.

*Note:* The “Get formats list” function cannot be used for playlists or YouTube channels. Also, it cannot be used for batches nor for more than one URL, iView episode or SBS episode.

*Note:* The formats choice overrides the “[Maximum resolution](#)” setting.

*Note:* the “Get formats list” setting is ignored if “Add URL to Batch” is chosen on the Min Dialog.

#### **“Force overwrites”**

By default, MacYTDL does not overwrite existing files. This setting causes MacYTDL to look for an existing file and if found, warns and gives the option to overwrite or rename the file or to return to the main menu.

By default MacYTDL will resume (i.e. continue) a partially complete download. This setting provides the option of ignoring partial or completed downloads, which are replaced with a new download.

*Note:* MacYTDL might not find a previously downloaded file and so might not provide a warning that it exists. It is best to assume that “Force overwrites” will replace an existing file without warning.

#### **“Use proxy”**

Some users need to supply a proxy URL in order to download a video. This setting passes the URL saved to *YT-DLP* which uses that URL to request the video for download. The URL must start with “http:”, “https” or “socks”. A URL must be entered to use the proxy feature.

#### **“Use cookies”**

Features on some web sites are only available if user credentials are other details are supplied. Sometimes, the only way to provide those details is via a “cookies” file. This setting passes the location of the user’s cookies file to *YT-DLP* which uses the cookies to log into and/or provide any required data to the web sites concerned. A valid cookies file must be provided using the drop down path control and must contain the required data.

*Note:* A cookies file can be created using extensions in the Firefox and Chrome web browsers. The file must be formatted as a “Netscape HTTP Cookie File”. It must be a text file with UTF-8 encoding.

#### **“Use custom template”**

Normally, MacYTDL gets the name of the video to be downloaded from the web site which serves the file. The custom file name template setting uses a name supplied by the user.

Beware: There is little error checking on the custom template. MacYTDL always checks that the last keyword in the template is the file extension. Otherwise, it passes the custom template setting on to *YT-DLP*. It is thus quite easy to cause errors. Also, to avoid overwriting files with the same name, using the same name for successive files will probably result in a failed download.

A custom template can contain virtually anything. There are two main types of template – which can be combined.

The first kind of template uses keywords recognised by *YT-DLP*. A list of keywords is available [in the YT-DLP readme](#). There are many keywords. Not all keywords apply to all downloads. Some, for example, only apply to music tracks.

There is a strict format which must be used for keywords to be recognised. They must be provided according to the [python string formatting operations](#). For example, all keyword entries must start with the ‘%’ symbol, be enclosed by brackets and followed by formatting operations. The most common operation flag is ‘s’. Thus the default template used by MacYTDL is “%(title)s.%(ext)s” – the video title followed by the file extension. With a few exceptions, any literal can be used between keywords and will be used in the file name. Note that some special characters cause errors in Windows and other systems.

A template with errors in formatting (e.g. a missing ‘%’) will usually be treated as a literal which will be used for the file name. A template with errors in content (e.g. “%(None)s” a non-existent keyword), will cause the default placeholder to be added to the file name. The default placeholder is ‘NA’.

The second type of template is a variation on the first which can control where files are saved. The ‘/’ symbol in a template is used to indicate the path of the saved download. The ‘~’ symbol indicates your “home” folder. Any missing folders will be automatically created. The standard path format on Windows systems can also be used. Some examples:

MyVideos/%(title)s.%(ext)s

– will save downloads in the “MyVideos” folder inside your download folder with the file name and extension used by the serving web site – it will create the “MyVideos” folder if it doesn’t exist.

~/Downloads/%(playlist)s/%(title)s.%(ext)s

– will save downloads with the file name and extension used by the serving web site in a folder named with the playlist name inside the “Downloads” folder inside the home folder. It will create a folder named with the playlist name if required.

~/Videos/%(series)s - my favourite/%(season)s/%(episode)s.%(ext)s

– will save downloaded videos with the file name taken from the episode title and extension used by the serving web site inside a folder named with the title of the season inside a folder named with the title of the series followed by the text “ - my favourite”, inside the Videos folder in the user’s home folder. Any missing folders will be automatically created.

/Music/%(title)s.%(ext)s

– The download will fail with the error: “unable to create directory [Errno 30] Read-only file system”. That is because of the ‘/’ character at the beginning of the template.

*Note:* It is possible for a custom template to result in a download file the name of which is too long for macOS to store. In that case, the custom template is ignored.

### **“Use custom settings”**

This setting enables any desired additional setting to be passed to *YT-DLP*. It is a potentially powerful facility as there is no limit on the number of additional settings. Custom settings are added to all the other settings specified in the “Settings” dialog and to any in the “[config](#)” file.

A common use of this setting is for storing credentials which MacYTDL currently cannot save. For example:

```
--username MyUserName --password MyPassword
```

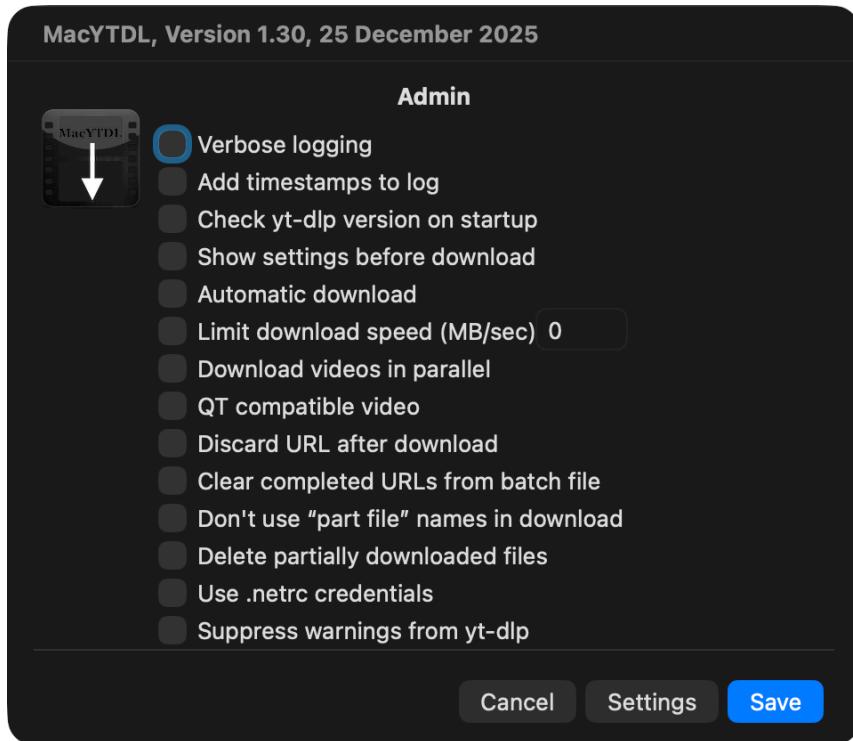
*Note:* MacYTDL does no error checking on the custom settings. If an invalid setting is used, *YT-DLP* will provide a report in the log file.

Detail on the range of settings available is provided here:

<https://github.com/yt-dlp/yt-dlp/blob/master/README.md>

## Admin

Various aspects of MacYTDL operation can be accessed from the “Admin” dialog. To access those controls, click on the “Admin” button in the Main dialog or press command-a. The Admin dialog will be displayed:



To save changes, click the “Save” button or press the Return key. That will save your changes and take you back to the Main dialog. To switch to the “Settings” dialog, click on the “Settings” button or press command-s. That will save changes and show the Settings dialog.

### ***“Cancel” – command-.***

Exit and return to the Main dialog without saving any changes. Keyboard shortcut is command-.

### ***“Settings” – command-s***

Saves changes to options and shows the “Settings” dialog. Keyboard shortcut is command-s

### ***“Save” – Return***

Save all changes to options and return to the Main dialog. Keyboard shortcut is Return.

### ***“Verbose logging”***

*YT-DLP* provide feedback on progress and what happened during the download. The feedback is stored in a file called “youtube-dl\_log-[the video name]-[date-time].txt” that is saved in this location: “~/Library/Preferences/MacYTDL”.

If there is a problem with the download, that file might provide clues on the cause. With the verbose option, *YT-DLP* provide more detail in the feedback. To open the folder containing the log files, use the “Open log folder” function in “Utilities”.

### ***“Add timestamps to log”***

This setting causes timestamps to be added to each line in the log files. Timestamps can help with diagnosing problems such as poor internet connections. The timestamps are in the format “yyyy-mm-dd hh:mm:ss”. This feature uses the “[ets](#)” tool which is stored within the MacYTDL bundle.

### ***“Check YT-DLP version on startup”***

When this setting is on, MacYTDL will automatically check whether a new version of *YT-DLP* is available every time MacYTDL is started. If so, the user is invited to update. This is useful as *YT-DLP* is updated regularly, sometimes every few days. Updates can improve downloading from some web sites, remove bugs from the *YT-DLP* core and add support for new sites.

### ***“Show download settings before starting download”***

If ticked, current download settings will be displayed with the option to download, change settings, cancel the download or quit MacYTDL.

*Note:* This setting is ignored if Automatic download is on.

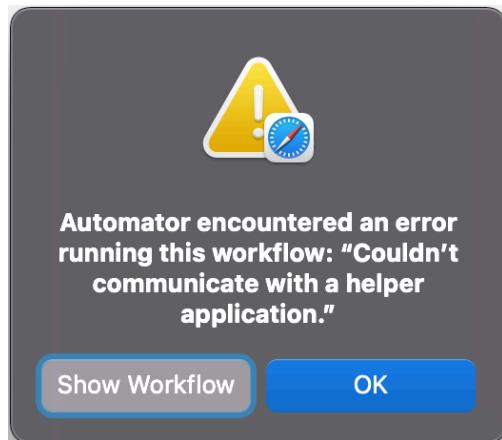


### ***“Automatic download”***

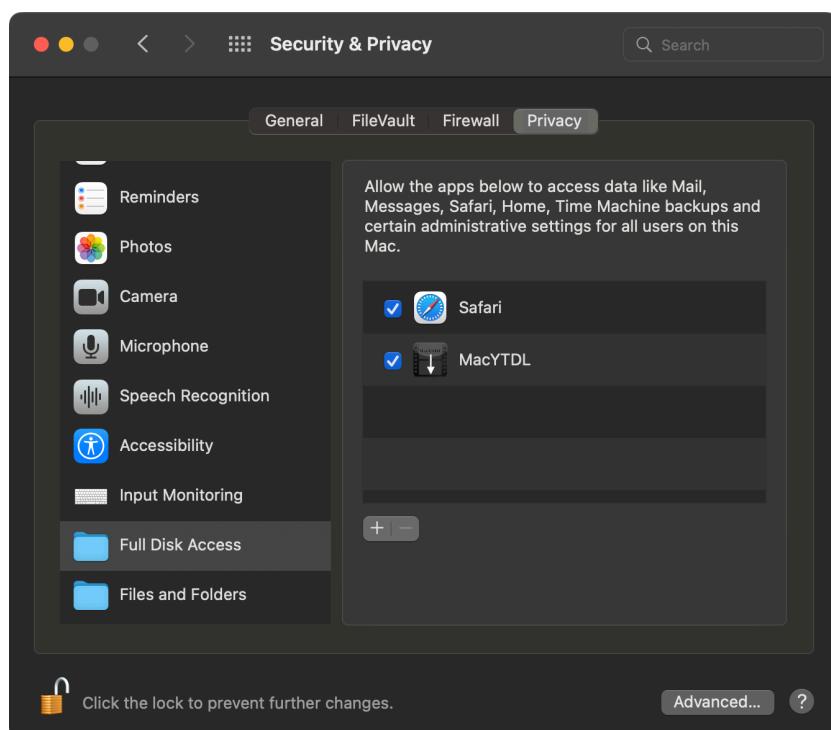
The Automatic download setting changes how the MacYTDL Service operates. With the setting off, the Service acquires the URL of the video to be downloaded, opens or switches to MacYTDL and pastes the URL into the URL text box. With this setting on, the Service will, after doing the usual checks, download the video immediately using the current download settings. This is a little quicker than switching backwards and forwards between the Main dialog and a web browser. Using this download method, it is not possible to set any one-time settings such as credentials (although, credentials can be specified in custom settings).

When this setting is on, for users of macOS 10.15 and newer, an option “Download Video Now” will show in the Services menu. When it is off, “Send-URL-To\_MacYTDL” will show in the Services menu.

This function is experimental. Generally, it does work. But, on some versions of macOS it can fail. In versions of macOS after macOS 10.15 Catalina, you will probably need to grant MacYTDL “Full Disk Access”. The application used to invoke the MacYTDL Service, e.g. Safari, should also be given full disk access. This is especially so if this dialog is shown:



To grant MacYTDL Full Disk Access, open System Preferences, open Security & Privacy, scroll down the left hand panel to “Full Disk Access” and use the right hand panel to add your copy of MacYTDL (NB. layout is different in macOS 13 Ventura and later versions):



*Note:* Automatic downloads are incompatible with the “Formats List” function. The latter setting is ignored if automatic download is being used (ie. from a web browser).

*Note:* A simpler facility is provided for choosing ABC iView episodes.

#### **“Limit download speed”**

This sets the maximum download speed. It is useful if download bandwidth is limited and other apps have a greater need. The limit rate is in megabytes/second and must be a real positive number e.g. “0.5” will limit download to 0.5 megabytes/sec.

*Note:* If it detects features it does not support, *YT-DLP* delegates downloading to *FFmpeg* (e.g. video delivered by HLS). In that case this setting has no effect.

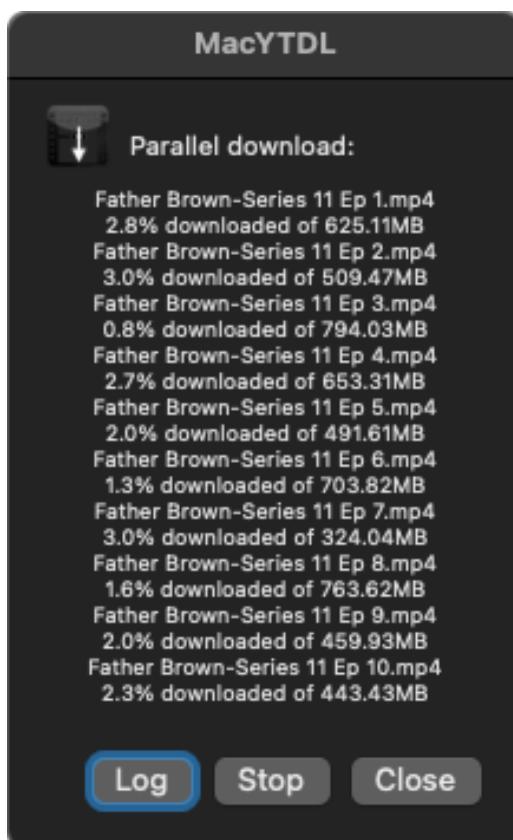
*Note:* A different speed can be set for each download. This setting does not control downloads which are underway.

#### **“Download videos in parallel”**

Many people have access to high speed broadband internet with speeds of 100Mbps and more. Some web sites, however, do not provide videos for download at such speeds. This setting enables multiple videos to be downloaded in parallel, separate processes. With parallel downloads, it is possible that downloads will use more of the available bandwidth thus giving a faster download of multiple videos. The improvement in time to download will vary with the size of the files, the web site serving the files and the speed of the Internet connection.

Parallel downloads can be used to download multiple videos (i.e. entering multiple URLs in the Main dialog), batches, iView episodes and playlists.

The Monitor dialog will show progress with all the downloads as in this example:



Clicking on the “Log” button will open the MacYTDL preferences folder in which log files are saved.

*Note:* This setting is similar to manually starting multiple separate downloads. However, there is one “Monitor” dialog instead of many. Nonetheless, each file is downloaded in a separate process and a log file is saved for each download file.

*Note:* There might not be much time saved if each download has to be processed by *FFmpeg* at the same time. But, that is less of an issue on fast Macs, especially those with “M” (ie. “Apple Silicon”) series processors.

*Note:* Downloads cannot be individually stopped. Clicking on the “Stop” button will stop all the downloads listed.

*Note:* Because of the amount of processing involved, the progress Monitor shows very little progress when a number of small files are being downloaded. Thus, for example, the “Stop” function might not be actionable before downloads are complete.

*Note:* The “Adviser” alert issued when downloads are finished does not have a “Play” function. The alert does, however, report how many files were downloaded and if any errors were found during the download.

### ***“QT compatible video”***

YouTube videos usually contain streams in the “VP9” video and “Opus” audio codecs in a “webm” container which cannot be played in Apple’s *Quicktime Player* nor in Quick Look. This setting addresses that issue. It involves re-encoding the audio and video streams in the download. The resulting file will have “H264” video and “AAC” audio inside an mp4 container.

Please note that because both audio and video are being re-encoded, it can take a long time to complete the process. The time taken will vary with the model of Mac being used, the length of the video, the resolution of the video stream and the sample rate of the audio amongst other factors. Using MacYTDL, conversions can take longer than in specialist tools such as *Handbrake*.

As a test, I found that it takes around 3 minutes to convert a 4 minute video on a Mac Studio M1 Max 24c. A slower Mac will take significantly longer. For example, a 2011 Mac Mini Core i7 took 127 minutes to convert a 17 minute 1920 × 1080 resolution YouTube video.

Turning on timestamps will help assess how long the conversion takes.

*Note:* The “Monitor” dialog can stop redisplaying and when hovered over that dialog, the mouse pointer shows the “spinning rainbow of death”. Research is continuing on this. However, it seems that it’s related to the Mac Sleep mode and/or when the display is off. To enable the Monitor to continue working, ensure that your Mac cannot go into Sleep Mode and the display stays on. If the Monitor does cease to work (ie. “hangs”), the only solution is to use *Activity Monitor* to quit the osascript process. Note, also, that although the Monitor can cease to function, the download and conversion to QT compatible format will continue to completion and the result video file will be playable in *Quicktime Player*.

*Note:* The conversion will be processed for files which are already Quicktime compatible.

### ***“Discard URL after download”***

This provides a choice on whether the URL used for a download should be retained in the Main dialog or discarded. Retaining the URL can be useful when testing different download settings.

### ***“Don’t use “part” file names in download”***

By default, *YT-DLP* uses temporary files with a “.part” extension during the download process. That makes clearer that the download is incomplete and facilitates continuing partial downloads. With this option, *YT-DLP* will not use the “.part” extension. The partial download file left from an incomplete download will be immediately playable.

*Note:* “part” files can be opened in video players. However, usually, it is necessary to remove “.part” from the file name.

*Note:* Turning on this setting can result in errors if the file already exists or can cause the download to be repeated and added to the existing file, doubling its size.

### ***“Delete partially downloaded files”***

By default, partially downloaded files are retained. With this setting on, partially downloaded and fragment files will be deleted.

*Note:* Downloaded subtitles files and thumbnail image files are never deleted.

*Note:* With this setting and “part” file also on, all “part” and fragment files are deleted if a download is stopped. However, if the “part” file setting is off, a playable partial download is usually retained (depending on the serving web site and the chosen download format) although it may not have audio.

### ***“Use .netrc credentials”***

Some web sites require credentials to be provided before downloading can commence. MacYTDL supports three ways for users to provide such credentials. The first is the “[Use custom settings](#)” function which can be used to provide a single username and password combination. The second is the “Credentials for download” option on the Main dialog which also accepts a single username and password combination but that combination is not saved. The third and perhaps most convenient, is by using a “.netrc” file. With a “.netrc” file, credentials for each web site can be saved in a plain text file.

The “.netrc” file must be stored in the user’s home folder and named: .netrc

i.e.: [Mac HD]/Users/[Home]/.netrc

*Note:* Because its name starts with a period, the “.netrc” file is hidden in Finder. To see hidden files press command-shift-. i.e. <command><shift><period>.

The format of credentials provided in the file for each extractor in the “.netrc” file is:

machine <extractor name> login <username> password <password>

So, for example, credentials to login to YouTube would be:

machine youtube login myaccount@gmail.com password my\_youtube\_password

*YT-DLP* has an “extractor” for each web site it supports. There is a [list of extractors here](#). A list of extractors can also be prepared using *YT-DLP* in Terminal:

```
yt-dlp --list-extractors
```

There is some more detail on the “.netrc” file [here](#).

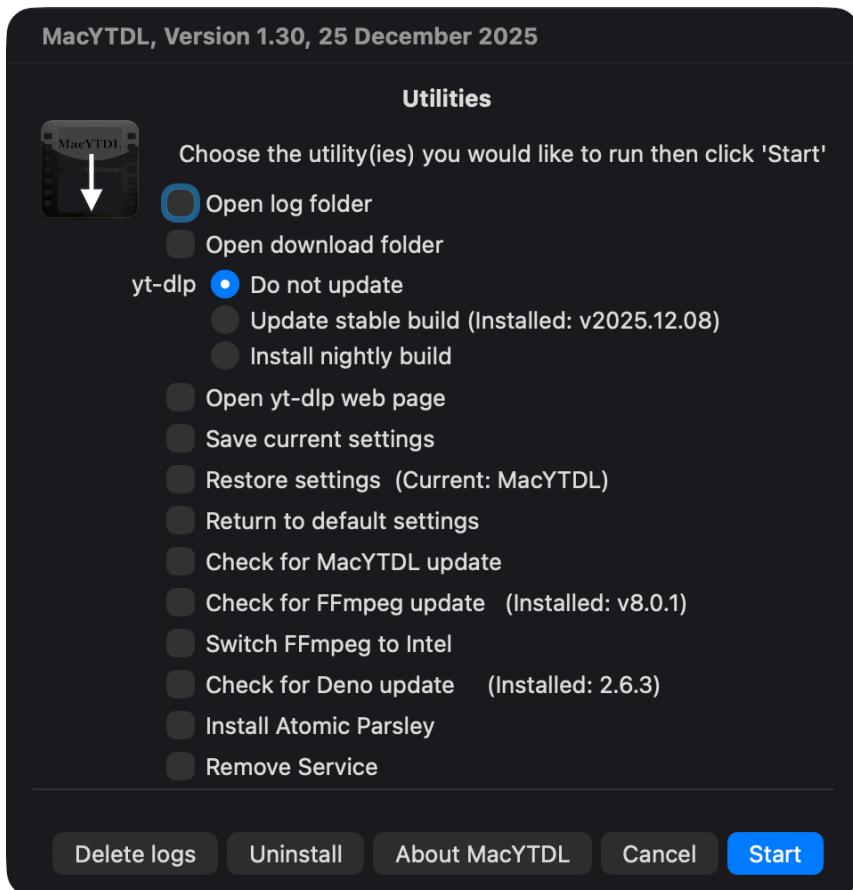
*Note:* “.netrc” files are not secure. It is sensible to consider protecting that file, for example, with strict file permission settings.

#### **“Suppress warnings from YT-DLP”**

Usually, MacYTDL will report any warnings found during its checking of a URL. With this setting, *YT-DLP* does not report any warnings during simulation and downloading.

## Utilities

A number of additional utilities are available to help with using MacYTDL. Click on the “Utilities” button or press command-u in the Main dialog:



Any combination of the twelve checkbox options can be selected at the same time. They will be performed in order when the “Start” button is clicked.

### ***“Open log folder”***

This will open the folder where all the *YT-DLP* log files are stored (in “/Users/[your home]/Preferences/MacYTDL/”). A separate log file is saved for each download.

### ***“Open download folder”***

This will open the folder where downloaded videos are saved.

### ***“Update YT-DLP”***

This function enables an update to the installed version of *YT-DLP* and switching between stable and nightly builds. The functions available depend on whether the current installed *YT-DLP* is a stable or nightly build.

Nightly builds of *YT-DLP* are now available. The nightly releases are the most up-to-date builds and include all changes made to *YT-DLP* to that point. There can be a number of nightly builds each week.

“Update yt-dlp stable build” checks whether a new stable build of *YT-DLP* has been released. If so, offers to download and install.

“Update yt-dlp nightly build” downloads and installs latest nightly build.

“Install yt-dlp stable build” offers to download and install the current stable build i.e. it switches from an installed nightly build to the current stable build.

“Install yt-dlp nightly build” downloads and installs latest nightly build i.e. it switches from the installed stable build to the latest nightly build.

*Note:* The version number shown is the version installed by the current user. Because *YT-DLP* can be installed by other users, the version shown can be out-of-date. Nonetheless, MacYTDL will only update a stable *YT-DLP* build if the installed version is not current.

#### **“Open YT-DLP web page”**

Opens the *YT-DLP* “Releases” web page in user’s default web browser. The Releases page provides detail on the changes made to *YT-DLP* in each release and the changes being prepared for the next release. The *YT-DLP* URL is: <https://github.com/yt-dlp/yt-dlp/releases>.

#### **“Switch to YT-DLP”**

This is visible only for users currently using *youtube-dl*. It switches MacYTDL from the *youtube-dl* script to *YT-DLP*.

#### **“Save current settings”**

This function enables a copy of the current settings to be saved. A “Save As” dialog is used. Settings can be saved anywhere although it’s best to use the default folder. Settings can be given any name but remain in “plist” format. It can be useful to give the saved settings file a name which indicates the nature of those settings e.g. “Youtube Music” or “TV Catch-up”.

*Note:* The “Save As” dialog may display improperly in macOS 10.13.6 High Sierra.

*Note:* This function copes poorly if a settings file is restored then deleted.

#### **“Restore settings”**

This function enables previously saved settings to be loaded and used in MacYTDL. Settings which are restored will replace all current settings. The name of the settings currently in use is shown beside this function. If no settings have been restored or settings have been returned to default, the name “MacYTDL” is shown.

*Note:* Trying to restore the MacYTDL.plist file will cause an error.

*Note:* I hope to redesign this function in the next MacYTDL release. It will be a “presets” function.

#### **“Return to default settings”**

This removes the current settings and returns all settings to the default. A warning is issued before going ahead to return settings to default.

#### **“Check for MacYTDL update”**

Checks whether a new version of MacYTDL has been released. If so, offers to download the new version. If downloaded, it will be saved into the MacYTDL downloads folder. Users can then drag the new version to the location they prefer.

### ***“Check for FFmpeg update”***

Checks whether a new version of *FFmpeg* and *FFprobe* has been released. If so, offers to download and install the new version.

*Note:* Users on macOS 10.12 and earlier cannot update the copy of *FFmpeg* installed by MacYTDL and so this function is hidden. Similarly, this function is hidden if a Homebrew version of *FFmpeg* is installed.

### ***“Switch to FFmpeg to [Intel/ARM]”***

Switches between the Intel and ARM64 versions of *FFmpeg* and *FFprobe*. By default, the Intel version of FFmpeg is used on Intel Macs and the ARM64 version on Apple Silicon Macs.

*Note:* Obviously, the ARM version is not available on Intel Macs.

### ***“[Install Deno/Check for Deno update]”***

Installs/Updates the *Deno* Javascript runtime. *Deno* is recommended by *YT-DLP* to download videos from YouTube if the *YT-DLP* version is more recent than 2025.10.22. It is optional in MacYTDL although downloads from YouTube are less likely to succeed without it. Updates to *Deno* are released regularly.

*Note:* A *Deno* install is also offered on the first occasion that a download from YouTube is started. However, it is only offered once and so is not offered on later occasions. It remains available in the “Utilities” dialog.

### ***“[Install/Remove] Atomic Parsley”***

*AtomicParsley* enables thumbnail images to be embedded in dowloaded files (e.g. album cover art). This function enables *AtomicParsley* to be installed if that was not done on the first startup. It enables *AtomicParsley* to be removed if it has been installed. It is used automatically by *youtube-dl*. *AtomicParsley* is not required by *YT-DLP*.

### ***“[Install/Remove] Service”***

This enables the MacYTDL Service to be installed if that was not done on the first startup. It enables the Service to be removed if it is installed. The MacYTDL Service takes a URL from the current text selection in any document, the Clipboard or the current web browser tab, switches to MacYTDL and pastes the URL in the MacYTDL URL text box.

### ***“Delete logs” – command-d***

Moves all the log files to Trash/Bin.

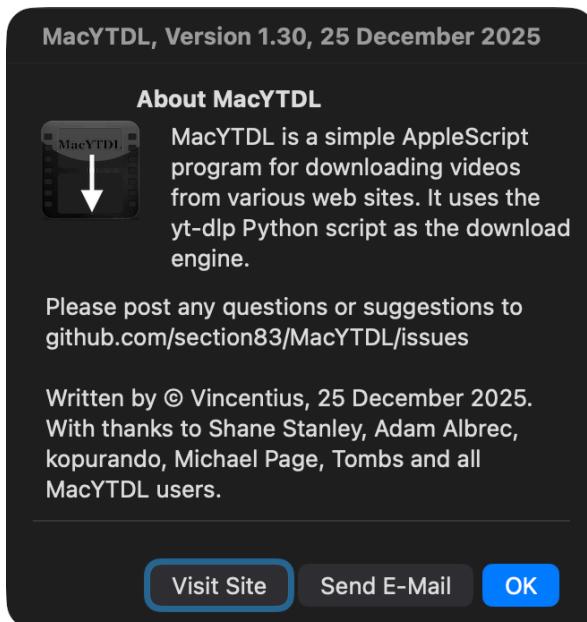
### ***“Uninstall” – command-shift-U***

Moves all files related to MacYTDL to the Trash/Bin. Asks for confirmation before moving files. Note that the Uninstall deletes all files including those belonging to other user accounts. If you change your mind you can put back all those files individually or put back just MacYTDL and run it again. Fresh copies of all components will then be downloaded and installed.

### ***“About MacYTDL” – command-a***

Displays a dialog with information about MacYTDL.

The “Visit Site” button opens the MacYTDL GitHub page in the user’s default web browser. The GitHub page provides some detail on MacYTDL, a facility for logging issues or suggestions and links to downloads of the current and old versions. The URL is: <https://github.com/section83/MacYTDL/>. The second button opens a blank message in the user’s default e-mail programme addressed to MacYTDL developer.



## Keystroke shortcuts

A number of keystroke shorts make using MacYTDL a little quicker:

### *Main dialog*

Return	Start download, Add URL to batch or Open batch functions as relevant
Command-u	Open Utilities dialog
Command-,	Open Settings dialog
Command-shift-,	open Admin dialog
Command-shift-/	Open this Help file in Preview
Command-q	Quit MacYTDL

### *Settings dialog*

Tab	Moves the selection between check boxes and function buttons
Space bar	Place a tick in the selected check box
Command-shift-,	Save changes and show the Admin dialog
Return	Save the settings
Command-,	Close and return to Main dialog without changing settings

### *Admin dialog*

Tab	Moves the selection between check boxes and function buttons
Space bar	Place a tick in the selected check box
Command-,	Save changes and show the Settings dialog
Return	Save the settings
Command-,	Close and return to Main dialog without changing settings

### *Utilities dialog*

Space bar	Place a tick in the selected check box
Tab	Moves the selection between check boxes and function buttons
Return	Start running the utilities checked
Command-d	Delete all log files
Command-shift-u	Uninstall MacYTDL
Command-a	Show the “About MacYTDL” dialog
Command-,	Close and return to Main dialog

### ***About MacYTDL dialog***

Return	Close and return to Main dialog
Command-v	Open the MacYTDL GitHub page and return to Main dialog
Command-e	Open a new message in user's default email app addressed to <a href="mailto:macytdl@gmail.com">macytdl@gmail.com</a>

### ***Settings shown before download***

Command-q	Quit MacYTDL
Command-e	Close and Open Settings dialog
Command-.	Cancel download and return to Main dialog
Command-d	Commence download

### ***Credentials dialog***

Return	Commence download (after entering credentials)
Command-r	Cancel download and return to Main dialog
Command-s	Skip credentials and commence download

### ***Batch functions dialog***

Command-r	Return to Main dialog
Command-e	Open the batch file for editing
Command-c	Clear the batch file i.e. remove all entries
Command-shift-u	Remove the last item in the batch file
Command-d	Commence downloading all URLs in batch file

### ***Monitor dialog***

Command-l	Open logs folder
Command-s	Stop download being monitored

## The MacYTDL Service – “Send-URL-To-MacYTDL”

MacYTDL includes an optional utility, the “Send-URL-To-MacYTDL” service. Services are small programs which provide extra functions in other applications. They are accessed using the “Services” item within the application and context menus.

Send-URL-To-MacYTDL works with any application that handles text. It makes downloading a video with MacYTDL a little quicker and easier. It takes any URL that has been copied or selected in any application and sends the URL to MacYTDL. If no URL has been copied or selected and a web browser is active, the Service will copy the URL of the current active tab and send that URL to MacYTDL. The web browsers in which it works are Safari, Brave, Opera, Firefox, Microsoft Edge, Chrome and Chromium.

The procedure to start with depends on where you find the video URL that you wish to download:

- If the URL is visible in text of any document in any application, select the text.
- If the URL is in a hyperlink in any document in any application, use the provided function to copy the link (e.g. in Safari, right-click the link text and choose “Copy Link”).
- If the URL is in the address bar of a currently active web browser, just go to the next step.
- Next, when you have copied or selected a URL or have that URL in your web browser:
  - go to the application menu (eg. the menu with the heading “Safari”);
  - browse down to “Services”;
  - choose “Send-URL-To-MacYTDL”.

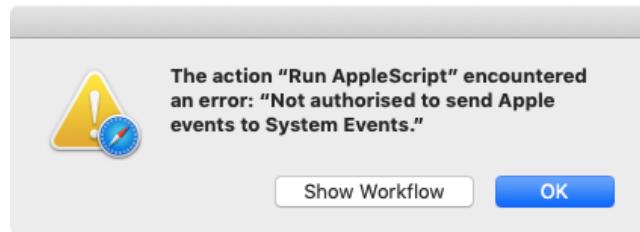
Send-URL-To-MacYTDL will launch or switch to MacYTDL and set the URL of the video to be downloaded.

If you don’t install Send-URL-To-MacYTDL when you first use MacYTDL, it can be installed later – see “[Utilities](#)” above.

Some people find using keystroke shortcuts to be quicker than using menus. In macOS 12 and later, to set a shortcut for the Send-URL-To-MacYTDL service:

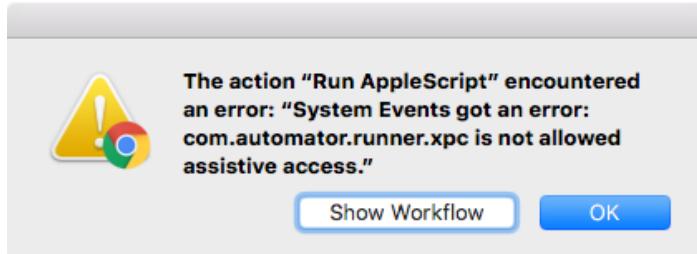
- open System Preferences;
- click on “Keyboard Shortcuts...”;
- click on “Services” in the left hand pane;
- click on “Services” in the left hand pane;
- scroll down to “General”;
- click in the tick box next to “Send-URL-To-MacYTDL”
- type the shortcut you wish to use.

On first using the Service, a message will probably be shown:



This means that the MacYTDL Service needs permission to use your web browser to send commands. In that case follow the procedure on [page 5](#) above to give Safari permission to control “System Events”.

Another message may be shown at the same time, if MacYTDL is running:

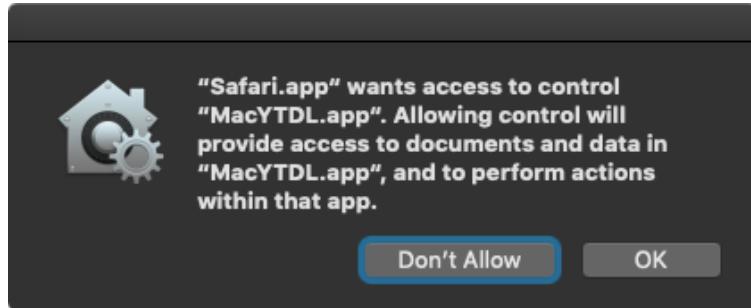


In that case, you will need to give permission for the MacYTDL Service to use your web browser to control your Mac:



- open “System Preferences”;
- click on “Security & Privacy”;
- click on “Accessibility”;
- click on the padlock and enter an administrator account credentials when asked;
- if your web browser is not shown, click on the “+” symbol and navigate to your web browser application (usually in the “Applications” folder) and click on “Open”;
- make sure there is a tick beside the web browser application;
- click on the padlock to prevent further changes.

If MacYTDL is not running, you may also see this message:



In that case, click on “OK” or follow the procedure on [page 5](#) above to give Safari permission to control MacYTDL.

The MacYTDL Service can also be used to automatically download videos instead of switch to/ opening the Main dialog. To use automatic downloads, turn on the “[Automatic download](#)” setting.

## Download log files

MacYTDL saves a log file for every download. The log files are created by *YT-DLP*. They contain detail on what happened during the download and progress with the download. They are therefore very useful when diagnosing failed downloads and other issues.

The log files are stored in: [Your Mac]/Users[Your Home]/Library/Preferences/MacYTDL/.

They are named:

ytdl\_log-[name of download file]-[date-time download initiated].txt

It is currently not possible to change that location. However, there is a function for deleting all log files in “[Utilities](#)”. There are also settings which change some of the log file contents in “[Settings](#)”.

All log files contain:

- estimate of download file size;
- download progress – as a percentage of estimated size (if that is known);
- download speed;
- estimated time to complete download as it progresses;
- download file name;
- some web site specific detail e.g. file ID on YouTube;
- details on warnings and errors encountered;
- if separate audio and video formats are downloaded, whether merging is done;
- name of original files deleted after merges or remuxes.

Additional detail is saved when the “[Verbose logging](#)” setting is on, including:

- echo of the settings used to generate the download including the URL e.g.  
`'--ignore-errors', '--newline', '--no-overwrites', '--verbose', '-o', '%(title)s.%s.(ext)s', 'https://www.youtube.com/watch?v=dBJbLV7xEhc';`
- version numbers for *YT-DLP*, Python, FFmpeg, and FFprobe;
- format specification sent to *YT-DLP* e.g. bestvideo\*+bestaudio/best;
- actual location of the download e.g. the “Google Video” or “Akamai” URL;
- echo of any FFmpeg command.

There is also a setting which add [timestamps](#) to each line of log files. Timestamps can be useful when diagnosing download problems.

## Using the default “config” file

*YT-DLP* can be controlled using a “config” file. The config file can contain additional settings and options which are not provided by MacYTDL. *YT-DLP* always checks for the presence of a config file and use its contents. Settings in the config file are added to those specified in the Settings dialog.

The config file must be a plain text file (e.g. created in TextEdit).

For *YT-DLP*, it may be stored in these locations:

- /etc/yt-dlp.conf [which will be used by all user accounts]
- Users/[Home]/.config/yt-dlp/config [which will be used by the “Home” user account]
- Users/[Home]/.config/yt-dlp.conf [which will be used by the “Home” user account]
- ~/yt-dlp.conf
- ~/yt-dlp.conf.txt

For *youtube-dl* it may be stored in these locations:

- /etc/youtube-dl.conf [which will be used by all user accounts]
- /Users/[Home]/.config/config [which will be used by the “Home” user account]

The layout rules for the config file are:

- Lines starting with a “#” are comments.
- Other lines are assumed to contain settings.

Some examples:

```
# Mark videos watched (currently only supported for YouTube)
--mark-watched
```

```
# Bypass geographic restriction via faking X-Forwarded-For HTTP header
--geo-bypass
```

```
# Download playlist videos in reverse order
--playlist-reverse
```

A future version of MacYTDL will provide for:

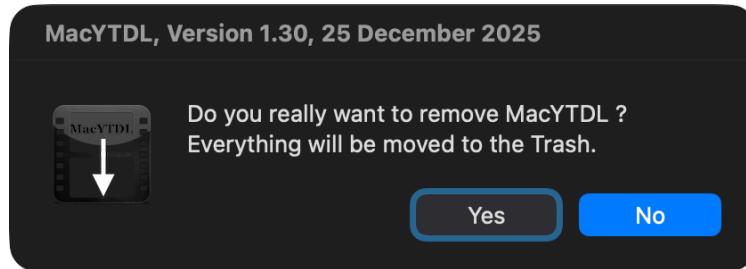
- specifying different locations for the config file; and
- multiple config files.

More detail on the config file is available here:

<https://github.com/yt-dlp/yt-dlp/blob/master/README.md#configuration>.

## How to uninstall MacYTDL

To uninstall MacYTDL, click on the “Utilities” button in the Main dialog and click on “Uninstall”. There is an opportunity to reconsider uninstalling:



Confirm that you wish to remove MacYTDL and all MacYTDL related files will be moved to the Trash/Bin. That will include all the log files created when downloading videos.

*Note:* The uninstall will remove all MacYTDL files for all user accounts. Files moved from each user's Library folder will be renamed with the user's account name as a prefix.

If you change your mind you can put back all those files individually or just put back MacYTDL and run it again. Fresh copies of all components will then be downloaded and installed.

*Note:* If you move files back into users' home folders, remember to remove the user's name.

## Using MacYTDL on versions of macOS prior to 12 Monterey

MacYTDL works on all Macs running macOS 10.10 and higher. However, a different version of the *YT-DLP* downloader is used on Macs running macOS 10.10 to 10.14. The two versions of *YT-DLP* are:

- for macOS 10.15 and higher – `yt-dlp_macos`;
- for macOS 10.10 to 10.14 – `yt-dlp_macos_legacy`.

The `yt-dlp_macos_legacy` version, however, cannot be updated past version 2025.08.11. That is because the *YT-DLP* developers can no longer support macOS prior to 10.15. Affected users might be able to install a more recent version of *YT-DLP* using *Homebrew* or *MacPorts*. It might also be possible to manually install all the *YT-DLP* dependencies and then manually install *YT-DLP*. More information is available from [the YT-DLP developers](#).

The installation of the correct version is handled by MacYTDL. However, the name of the installed downloader is always “`yt-dlp`”.

Users currently using *youtube-dl* can continue to do so. However, MacYTDL no longer offers the option of updating *youtube-dl* because it has not had a stable release since December 2021. An option to switch from *youtube-dl* to *yt-dlp* is available in the Utilities dialog.

There are also limitations on the use of FFmpeg for users on a range of macOS versions:

- Users on 10.13, 10.14, 10.15 and 11 cannot use the default *FFmpeg* install as it is compiled for macOS 12+.
- Users on 10.10, 10.11 and 10.12 cannot use *FFmpeg* after v6.0 (because they lack Metal API & Core Image is located inside Quartz.framework).

## **YouTube URLs and playlists/channels**

There are a number of formats for YouTube URLs. The different formats lead to different results in MacYTDL. They can also result in the download of a very large number of files. MacYTDL pops a warning if a playlist contains more than 20 items.

The playlist items downloaded can be set by adding a playlist item specification to “Use custom settings” in “Settings”. The format of the specification is:

--playlist-items [item spec]

For example with the specification “--playlist-items 1:3”, MacYTDL will download the first three items of the playlist. Detail on the [item spec] format is provided in the first item here:

<https://github.com/yt-dlp/yt-dlp/blob/master/README.md#video-selection>.

This is a list of the common formats and what happens in MacYTDL.

YouTube playlists can be downloaded in [parallel](#).

### ***Single video***

<https://www.youtube.com/watch?v=zNpM6-5pFF4>

<https://youtu.be/YlRY3KQpib4>

A link to a single video or an abbreviated link to a single video. Individual videos usually have “watch?v=” in the URL or have no other identifying elements (as in the abbreviated form). MacYTDL will download the video.

### ***A playlist***

<https://www.youtube.com/playlist?list=PLA2-nXsHvm3LGeixJDJ29HT1saHhgLHHe>

This is a YouTube playlist. Playlist URLs often contain “playlist?list=”. The keyword “list” usually points to a particular playlist. Another indicator that this is a playlist URL is the “PL” at the beginning of the playlist ID.

MacYTDL will download all items in the list. It will warn if there are more than 20 items in the list and will advise if there were errors in the download.

*Note:* Playlist items are downloaded sequentially. It can take a long time to setup the download and then a long time to download the entire playlist.

### ***Single playlist item***

<https://www.youtube.com/watch?v=zNpM6-5pFF4&list=PLA2-nXsHvm3LGeixJDJ29HT1saHhgLHHe&index=1>

This is a single playlist item. These URLs typically end with “&index=”. MacYTDL will download just that item.

*Note:* To download the entire playlist, remove the “&index=[n]” part of the URL.

### ***A playlist item***

<https://www.youtube.com/watch?v=zNpM6-5pFF4&list=PLA2-nXsHvm3LGeixJDJ29HT1saHhgLHHe>

[https://youtu.be/X0WOa\\_vDUFA?list=PLF76BA379BC905431](https://youtu.be/X0WOa_vDUFA?list=PLF76BA379BC905431)

This is a playlist URL, or an abbreviated URL, that opens on a single item which usually starts playing automatically in a web browser. MacYTDL will download all items in the playlist. It will warn if there are more than 35 items in the list and will advise if there were errors in the download.

### **A channel**

<https://www.youtube.com/user/ThePoguesOfficial>

There are a number of formats for YouTube channel URLs. Some URLs are in a custom format. There can be a great many videos available. If there are more than 20 items, it will show a warning before going ahead with the download. Currently downloading from channels with *youtube-dl* has been disabled. This will be reviewed prior to the next release of MacYTDL.

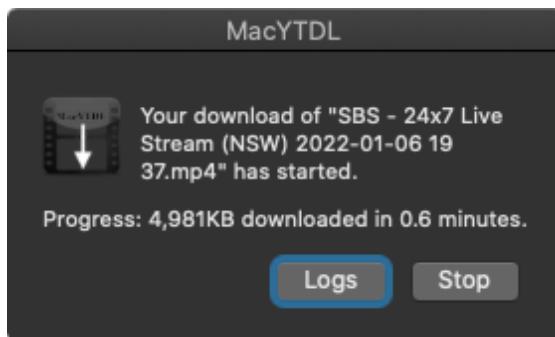
*Note:* Channel items are downloaded sequentially. It can take a long time to setup the download and then a long time to download all the videos in a channel.

### **Live stream**

<https://www.youtube.com/watch?v=W1ilCy6XrmI>

Although it looks very much like a single video URL this is a live video stream URL. As yet, I have not found a reliable identifying feature in stream URLs. MacYTDL will download the stream until the “Stop” button is pressed on the Monitor.

The Monitor dialog can only be closed by stopping the stream download. This is to prevent download processes being left running with no facility to close them down.



*Note:* The recording will start from the moment MacYTDL starts to download the stream not from the beginning of the stream (which might have been some hours/days earlier).

*Note:* Some streams cannot be identified as such and so the Monitor dialog will show the usual “Close” button. To ensure such a stream download can be stopped avoid clicking on “Close”.

### **An uploader playlist**

<https://www.youtube.com/playlist?list=UUBABnxM4Ar9ten8Mdjj1j0Q>

This is a playlist of videos from a particular uploader (i.e. a person or an organisation). Typically, the playlist ID starts with “UU”. MacYTDL will download all items in the playlist. It will warn if there are more than 35 items in the list and will advise if there were errors in the download.

*Beware:* Some uploader playlists have a great many items. The example above has over 1,100 items. Such playlists take a very long time to prepare and then download.

## **A YouTube mix**

[https://www.youtube.com/watch?  
v=NrLkTZrPZA4&list=RDEMtu8TSn01ATwUIqXnxfaf6zQ&start\\_radio=1](https://www.youtube.com/watch?v=NrLkTZrPZA4&list=RDEMtu8TSn01ATwUIqXnxfaf6zQ&start_radio=1)

[https://www.youtube.com/watch?v=8UVNT4wvIGY&list=RD8UVNT4wvIGY&start\\_radio=1](https://www.youtube.com/watch?v=8UVNT4wvIGY&list=RD8UVNT4wvIGY&start_radio=1)

A YouTube mix is a playlist that has been curated by YouTube. Mixes are formed using the viewing history of each user (if known to YouTube). MacYTDL will download all items in the playlist. It will warn if there are more than 35 items in the list and will advise if there were errors in the download.

*Beware:* YouTube mixes can have a lot of items and so take a long time to prepare and then download.

## **Personal “Watch Later” playlist**

<https://www.youtube.com/playlist?list=WL>

A “Watch Later” playlist is very handy. Downloading, however, is not easy at present. I have not been able to get *youtube-dl* to download my “Watch Later” list by supplying my credentials directly. I have succeeded only by supplying my credentials in a “cookies” text file ([see above](#)). This will download all your YouTube “Watch Later” videos into your home folder. Remember to use ordinary double quotes, not the curly “smart” quotes you get in word processors. I have not been able to test the “Watch Later” downloads with *YT-DLP*. MacYTDL will warn if there are more than 35 items in the list and will advise if there were errors in the download.

*Note:* The cookies file must be encoded with UTF-8. Most cookies utilities will use that encoding.

## Downloading videos from ABC iView (in Australia)

ABC iView in Australia serves most individual videos from what can be called “show” pages. Episodes are not available from their own web page. For example, all episodes for all seasons of the show *Back Roads* are available on the one “*Back Roads*” show page.

*Note:* This feature is not available to users on OS X 10.10 (Yosemite), 10.11 (El Capitan) and 10.12 (Sierra) because the version of the “curl” binary supplied does not present the correct certificates to the iView web site.

If the URL of a “show” page is pasted into MacYTDL and there is more than one episode, a separate dialog will be displayed listing all the episodes available. The desired videos can then be chosen for download. There is an “All episodes” checkbox which will download all episodes. This is an example iView “Chooser” dialog:



MacYTDL will also locate and list any “Extras” videos that are available for a show.

On iView the ABC has some shows for which there are a great many episodes (“Gardening Australia has 128). MacYTDL will try to show all the episodes in this dialog. On a Mac with low screen resolution (e.g. 1280x720), however, the dialog might be too large to show all episodes if there are more than 80. Hopefully a future version of MacYTDL will address that problem.

Notice that for some shows, episodes are shown in reverse order to match the order mostly used in iView.

At present MacYTDL can download from only one show page in each download process. However it can download:

- from any number of show pages if each download is requested separately;
- any number of episodes from any series of a single show; and
- any number of separate episodes from any show pages in one process if the URLs of each episode are supplied manually.
- To download a number of individual episodes from more than one show page:

- navigate to the particular episode desired;
- click on the episode's thumbnail image to start the iView player;
- copy the URL from the address bar;
- switch to MacYTDL;
- paste the URL then add a space;
- return to iView, navigate to the new desired show episode and repeat the above steps.

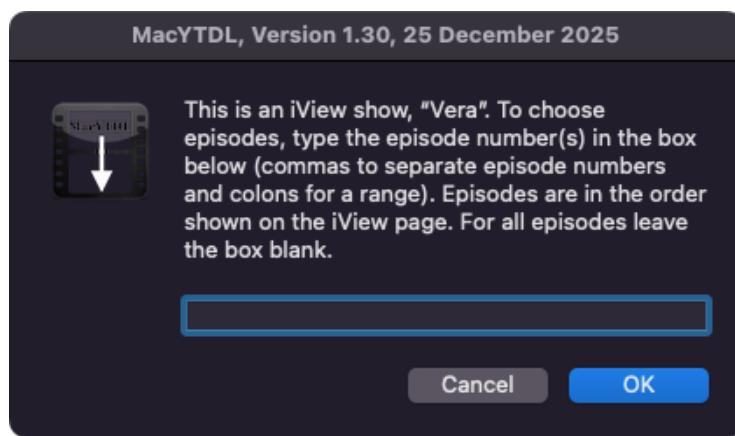
If a show page has only one episode, MacYTDL will automatically commence to download that episode.

A separate destination folder can be specified in the “One-time settings” on the Main dialog. However, that must be done before selecting “Continue”. A new download folder can also be requested by using a custom file name template ([see above](#)).

MacYTDL can also download and ‘record’ live streams from the ABC iView web site. It works in much the same way as for YouTube streams: MacYTDL identifies the download as an ABC live stream and shows the correct form of the Monitor. MacYTDL will download the iView stream until the “Stop” button is pressed on the Monitor.

#### ***Automatic download – Choosing iView episodes***

The above method of choosing iView episodes is not available when using [Automatic download](#). When automatic download is invoked, if the front most web page is showing a season of a show page, this dialog will be shown:



To download all episodes for the visible season, press “OK”.

To download a selection of episodes, type their numbers in the box. Note that episodes are numbered in the order shown on the iView web page. Usually, iView shows episodes with the most recent on the left. Thus, to download the newest episode type a “1”. Multiple episodes can be downloaded. Use commas to separate individual episodes (eg. 1,2,4) and colons to indicate a range (eg. 5:9).

*Note:* iView episodes cannot be downloaded in parallel when requested using the automatic download function.

## **Working with dual monitors**

By default MacYTDL shows the Main dialog on the main screen of a dual monitor setup. If it is moved to the secondary monitor and closed, it should be displayed in that new location when reopened. However, results depend on the particular setup. The Monitor dialogs are always displayed in the top right hand corner of the main display.

Problems can arise if the screen layout is changed while MacYTDL is running. This is because the position of MacYTDL's dialogs is taken from the previously saved coordinates which become invalid if the layout is changed. It's best not to change the layout while MacYTDL is running. Making MacYTDL more friendly with dual monitor setups is still a work-in-progress.

## Batch functions

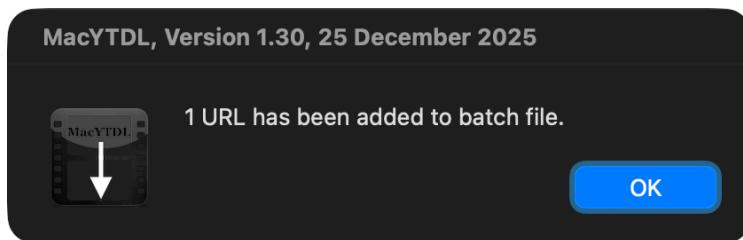
There can be situations in which you want to download a number of videos at another time. For example, download speeds can be faster outside peak hour periods. The MacYTDL “Batch” functions enable URLs to be saved and downloaded later. URLs are stored in a file named “BatchFile.txt” which is located in:

/Users/[Home]/Library/Preferences/MacYTDL/

To add a URL to the batch file:

- in the Main dialog, click on the “Add URL to Batch” checkbox;
- paste the URL into the text box (if you did not use the MacYTDL Service); and
- click on the “Continue” button.

The URL will be checked to ensure it is valid then, added to the batch file and a message displayed accordingly. This involves checking with the video web site concerned and so can take time.



Multiple URLs can be added in one step. URLs need to be separated by either spaces, tabs, returns or linefeeds. Thus, a list of URLs can be copied from a source document and added to the batch in one step.

The current “Remux format” setting and the expected file name are also saved in “BatchFile.txt”. That is so the “Adviser” can play the first video in the batch when download is complete. Other settings are respected at the time of download. However, the “Get formats list” setting is not used is ignored when a batch is downloaded.

*Note:* This is an interim arrangement. When time permits a better batch download function will be developed which is more accurate and flexible.

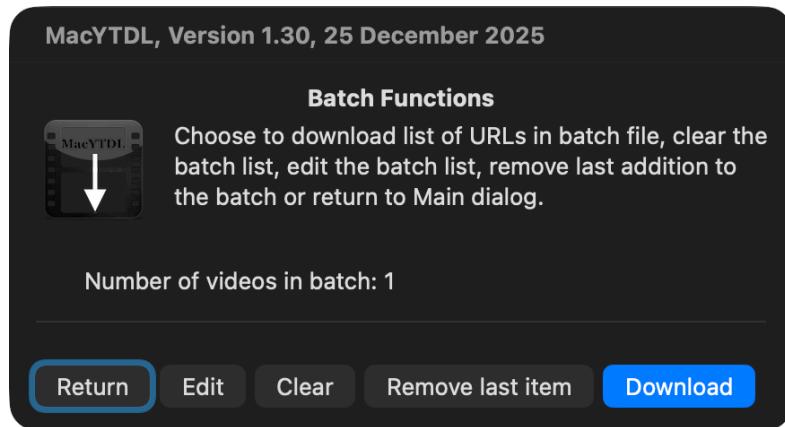
*Note:* Download files which have “#” characters in their names must be avoided. Any downloads added to the batch which include the “#” in the file name will cause an error. This is because it causes confusion with the “#” comment delimiter used by *youtube-dl* and *YT-DLP* in batch files. This issue will be corrected in the future revamp of batch processing.

The “Add URL to Batch” checkbox will remain ticked until manually unticked or the user quits MacYTDL.

Any number of URLs can be added to the batch file in one step by pasting the URLs separated by spaces.

Batches can be downloaded in [parallel](#).

A separate dialog is provided to handle all other batch functions. To display the Batch functions dialog, in the Main dialog, click on the “Open Batch functions” checkbox then click on the “Continue” button. This dialog will be displayed:



To download all the videos saved in the batch file, click on the “Download” button. The downloads will commence and continue, one at a time or in parallel depending on the current setting.

*Note:* All videos in a batch will be downloaded using the same settings. It is not possible to have different settings, e.g. maximum resolution, for each video in a batch.

There is a setting in the “Admin” dialog to control what happens when a batch download is complete, called “Clear completed URLs from batch file”. When turned on, URLs which were downloaded without error are removed from the batch file. When turned off, all URLs, both those downloaded with errors and those without, remain in the batch file.

Log files saved for batch downloads are always named “yt\_dl\_log-Batch\_download\_on-[date-time]”. They contain detail on progress with the download including on any errors encountered. However, if parallel downloads are requested, a separate log file is saved for each download, named in the standard way.

Clicking on the “Edit” button will open the batch file in the user’s default text editor. The batch file is a plain text file. There are restrictions on editing the file:

- the file must be saved as a plain text file – ie. not as rich text or word processing file;
- each line must contain a URL followed by “#” then the correct file name for the file to be downloaded;
  - the end of the file name must be marked by a “\$” symbol;
- there can only be one URL per line; and
- there must be a blank line at the end of the file – there can be no other blank lines.

The “Clear” button will remove all saved URLs from the batch file.

The “Remove last item” button will delete the most recently added URL from the batch. That can be useful if a URL was added by accident.

The “Return” button closes the Batch Functions dialog and returns to the Main dialog.

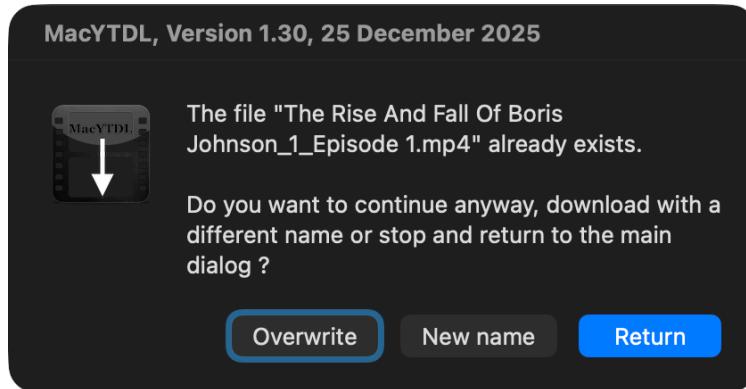
*Note:* Beware of including live stream URLs in a batch. That is because live streams continue recording until manually stopped with the Monitor dialog. No other downloads in the batch will proceed while a live stream is being downloaded.

There are a number of limitations with this batch function:

- Download settings apply to all videos in a batch – ie. all settings, except remux, are not saved with each URL. Thus, downloads are controlled by the settings current at the time of download. So, if a setting such as “Audio only” is set at download time, then only the audio component of each video will be downloaded. It is usually safer to store URLs in the batch file that are all for video files or all audio files rather than mixing them up.
  - Consequently, all videos in a batch will be saved in the same downloads folder although this behaviour can be changed with the custom file name template setting ([see above](#)).
- Invalid text in the batch file may cause *YT-DLP* to post an error. MacYTDL will ask *YT-DLP* to ignore errors and continue downloading but might not advise of that error. In any event the error should be reported in the log file.
- Some videos from ABC Australia’s iView trigger a warning from *YT-DLP* regarding subtitles. Those videos are nonetheless added to the batch and can be downloaded. However, they cannot be played by the “Adviser”.
- Incomplete downloads are not automatically deleted by the Monitor dialog “Stop” function.
- MacYTDL does not check for duplicates in the batch file. *YT-DLP* will download all files in the batch but will not save any duplicated files unless “%(autonumber)s” is added to the [custom template setting](#) at download time.
- It is not possible to choose which videos in a batch to download. However, lists of URLs can be saved in separate files and copied and pasted into the batch file for downloading.
- The “Play” function available when downloads are completed will not work if a file name manually added to the batch file is incorrect.

## Overwriting existing videos ?

By default, MacYTDL will not overwrite an existing file. If the “[Force overwrites](#)” setting is on, it checks whether a video with the same name has been saved in the downloads folder. If so, a warning dialog might be shown:



*Note:* MacYTDL does not warn of partially downloaded files. If “Force overwrites” is on, those files will be replaced. Sometimes, MacYTDL does not find the previously completed download. That can be because the file name was changed, for example, by a remux. Nonetheless, The previous file will be replaced.

### ***“Overwrite”***

MacYTDL will move the existing file to Trash/Bin and commence downloading the new file.

### ***“New name”***

MacYTDL will commence downloading the new file with “-2” appended to the file name.

NB. At present, if there is more than one file to be downloaded, e.g. a playlist, ABC iView episodes, a batch or multiple video download, MacYTDL will add “-2” to the names of all files downloaded, not just those which already exist. That is because although *YT-DLP* can download a number of files in one process, it apply only one output template for all the file names.

### ***“Return”***

Downloading is cancelled and the Main dialog is displayed.

## Using Homebrew and MacPorts with MacYTDL

MacYTDL can use Homebrew and MacPorts installs of *YT-DLP* and *FFmpeg*. It will recognise installs on both Intel and Apple Silicon Macs.

If there is a need to switch from a Homebrew install to the MacYTDL supplied executables, the following should work:

For Homebrew installs of *YT-DLP*:

- On Intel Macs, delete or rename this symlink: /usr/local/bin/yt-dlp.
- On ARM Macs, delete or rename this symlink: /opt/homebrew/bin/yt-dlp.

For Homebrew installs of *FFmpeg*:

- On Intel Macs, delete or rename these symlinks: /usr/local/bin/FFmpeg and /opt/homebrew/bin/ffprobe.
- On ARM Macs, delete or rename these symlinks: /opt/homebrew/bin/FFmpeg and /opt/homebrew/bin/ffprobe.

For MacPorts installs of *YT-DLP*:

- On all Macs, delete or rename this symlink: /opt/local/bin/yt-dlp.

For MacPorts installs of *FFmpeg*:

- On all Macs, delete or rename these symlinks: /opt/local/bin/ffmpeg and /opt/local/bin/ffprobe.

When MacYTDL is next started, it will prompt to installing the missing executable(s).

Users should note, however, that:

- A Homebrew/MacPorts install involves significantly more files – all dependencies including Python are installed separately. The *YT-DLP* and *FFmpeg* executables installed by MacYTDL have all dependencies and Python as needed built-in.
- MacYTDL cannot update a Homebrew/MacPorts install.
  - Thus, the “Check *YT-DLP* version on startup” and “Check for FFmpeg update” functions are hidden.
- A MacYTDL install by default is preferred and any Homebrew/MacPorts install then ignored.
- To use a Homebrew install with MacYTDL:
  - On Apple Silicon Macs, the executable file “/usr/local/bin/yt-dlp” must be deleted, if present.
  - On Intel Macs, the executable file “/usr/local/bin/yt-dlp” must be deleted before running the Homebrew install. This is because Homebrew, by default, stores an alias to the *YT-DLP* executable in the same place as MacYTDL and Homebrew will not overwrite an existing executable.

- Similarly, to use a MacPorts install with MacYTDL, on all Macs, the file “/usr/local/bin/yt-dlp” must be deleted.
- MacYTDL has no facility for handling a Homebrew/MacPorts install.
- MacYTDL has no facility for switching from a Homebrew/MacPorts install to a MacYTDL install.
- MacYTDL assumes that the *YT-DLP* executable is always called “*yt-dlp*”. Any other name will not be recognised.

Testing done so far suggests that Homebrew formulae and MacPorts ports sometimes install versions which are older than the current versions, especially FFmpeg.

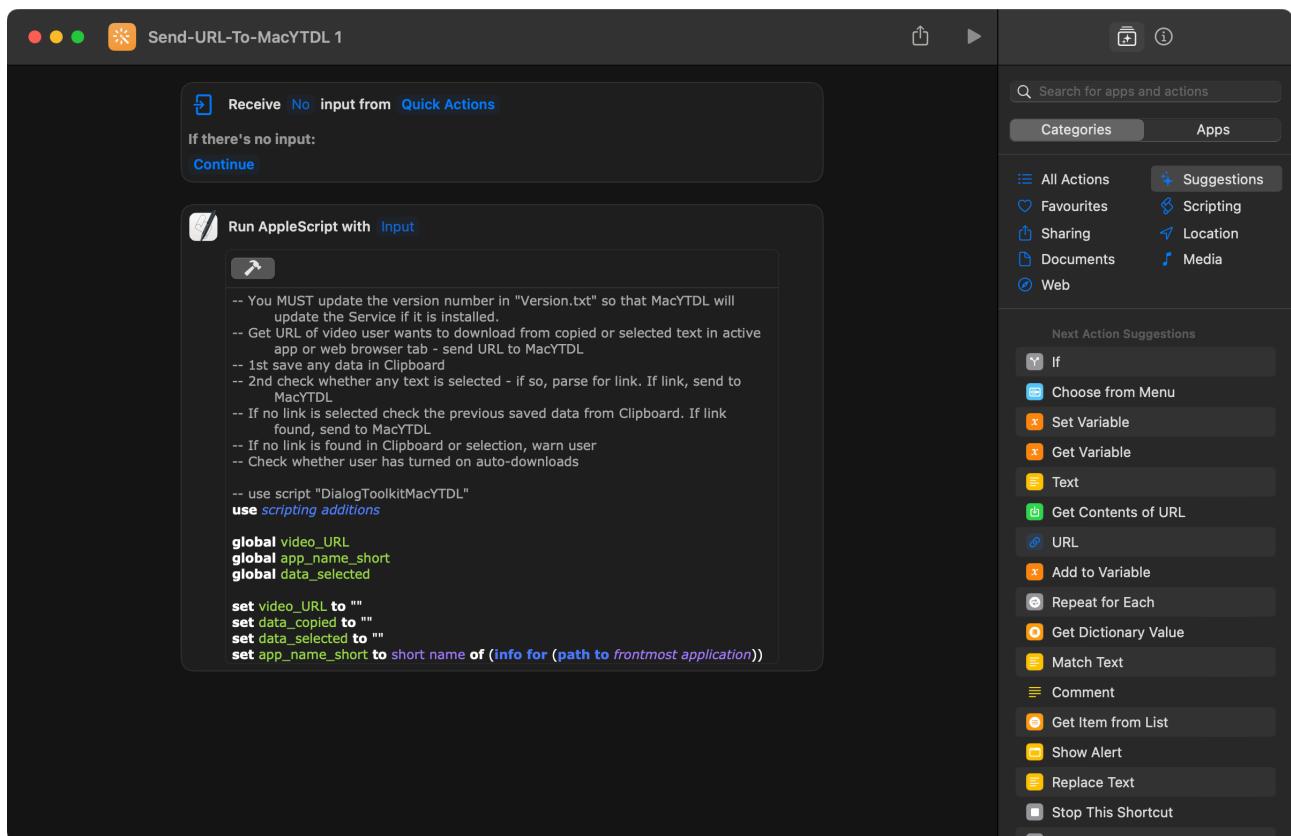
## “Shortcuts”

[Apple explains](#) that Shortcuts provide “a quick way to get things done with your apps, with just a click or by asking Siri.”. They have been available in macOS since macOS 12 Monterey.

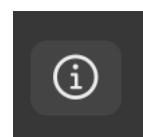
Shortcuts are quite similar to Automator workflows. In fact, the company that developed Shortcuts called it “Workflow”. There are no Shortcuts available in MacYTDL. However, the MacYTDL Service, “Send-URL-To-MacYTDL”, can be readily converted into a shortcut. It can then be invoked from more places. In testing so far, it seems faster and more stable than the Automator Service.

To convert the MacYTDL Service into a shortcut:

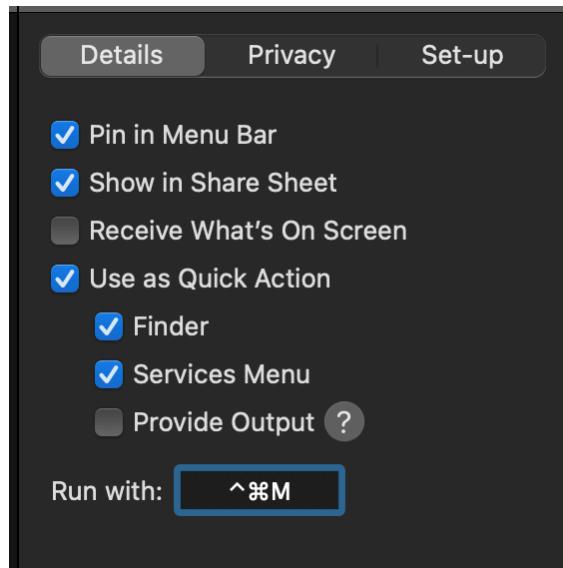
- Make sure the Service is installed. In MacYTDL, open “Utilities” and, if available, choose “Install Service”.
- Navigate to:  
[Your Mac]/Users/[Your home]/Library/Services
- Right-click the file “Send-URL-To-MacYTDL”.
- Choose “Open With”.
- Choose “Shortcuts”.
- The Shortcuts app should open and look like this:



- Click on the “i” info button in the top-right hand corner of the window:



- Choose how you would like to access the shortcut. I have chosen to access from many places.
- If desired, choose a Keyboard Shortcut. I use control-command-m:

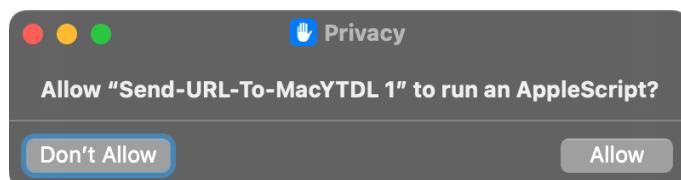


You can also add the shortcut to the Dock:

- Choose the “File” menu.
- Choose “Add to Dock”.

When you have finished setting up the shortcut, you can quit the Shortcuts app or just click the close button on the Send-URL-To-MacYTDL shortcut window.

A permission dialog will be shown on the first occasion the Shortcut is invoked:



Click on “Allow”.

*Note:* If you choose “Add to Dock”, another file named “Send-URL-To-MacYTDL” is created and saved in “[Your Mac]/Users/[Your home]/Applications”. Also, that file remains in place if the Shortcut is deleted in the “Shortcuts” app or removed from the Dock.

## **What if the download doesn't work ? Troubleshooting**

There are many reasons why a download might fail or not produce the desired result. MacYTDL does do a “simulated” test download and reports errors found (eg. because of invalid URLs or DRM limitations). But, that test cannot find all possible problems.

There is an option to attempt to download even when there is an error as well as some warnings. The download might well fail. If so, check the log file for possible reasons.

MacYTDL will show a warning that there was a problem if the download was not 100% complete or an error was reported by *YT-DLP*. Also, if a download is interrupted, it is possible for *YT-DLP* to continue the download from the point of interruption or to download a missing component. So, it is worthwhile trying the download again.

*Note:* An error with a video in a YouTube playlist will not prevent other videos in the list from downloading. If an error message is displayed, just click on “Download” to download the available videos.

A common problem is that a web site has changed and *YT-DLP* is no longer able to find the correct file to download. The best solution is to check regularly for and install updates to *youtube-dl/YT-DLP*. Use the “Check for YT-DLP update” function or turn on the “Check YT-DLP version on startup” option.

Another common issue is AES-SAMPLE encryption or another DRM control. Many videos are encrypted with DRM protection and cannot be downloaded by *YT-DLP*. That is a matter of principle.

Sometimes, issues are caught when MacYTDL checks the URL. In that case, what happened is noted in this file:

/Users/[Home]/Preferences/MacYTDL/youtube-dl\_simulate.txt

Note that the simulate file is overwritten by the next download.

If MacYTDL proceeds to download the video, *YT-DLP* produces a detailed log. If the download fails, that log might help to diagnose problems. MacYTDL saves the *youtube-dl* log in a separate file for each download. The file is saved here:

/Users/[Home]/Preferences/MacYTDL/youtube-dl\_log-[name of video].txt

The “verbose” option will provide more detail to help diagnose issues.

Information on how *YT-DLP* functions is available at:

<https://github.com/yt-dlp/yt-dlp/blob/master/README.md>.

### ***What if internet connection is lost during a download ?***

*YT-DLP* will pause all downloading if a connection is lost. The MacYTDL Monitor remains visible showing that time has elapsed but that there has been no further download. *YT-DLP* will continue the download(s) after the connection is restored.

## Limitations and known bugs

MacYTDL has some limitations:

- Cannot download encrypted or DRM protected videos – this is a limitation of *YT-DLP* imposed as a matter of principle.
- Does not use options to overcome geo-blocking.
  - However, “[config](#)” file and “[custom settings](#)” can be used to supply desired geo-blocking settings.
- Does not provide access to some *YT-DLP* settings (eg. video password, retries, dates, archives, filters). *YT-DLP* has many features which MacYTDL cannot access at this time. The “[config](#)” file and “[custom settings](#)” settings provide ways to use those features.
- Provides little control over the format of videos downloaded (eg. can’t download and merge a video-only file with a video+audio file which enables higher quality audio with some downloads). Again, the “[config](#)” file and “[custom settings](#)” facilities provide ways to access those features.
- The Send-URL-To-MacYTDL service cannot work when the web browser is in full screen mode. Therefore, it takes the browser out of full screen before switching to MacYTDL.
- It is not possible to download videos with the same name into the same folder unless the “% (autonumber)s” parameter is used in the custom file name template ([see above](#)).
- A duplicate download in a batch is not saved to disk (protection provided by *YT-DLP*).
- The “[SponsorBlock](#)” functions of *YT-DLP* have not been implemented. Users can, however, use the [custom settings](#) facility in MacYTDL to implement sponsor blocks.

MacYTDL also has some bugs:

- Starting with macOS 12.3, MacYTDL can produce permission and file does not exist errors. Work on solving those is continuing.
- Dialogs can appear in unpredictable locations when layout of dual screen setup is changed.
- The Monitor dialog and the download complete notification sometimes have the wrong extension on the file name shown (e.g. when a remux has been specified).
- The “Stop” function does not always delete incomplete downloads (including when using a custom file name template and batch downloads).
- On occasion, alerts advising on installation progress do not automatically close.
- A warning that the download already exists might be issued even though a recode to a different format has been specified.
- “Monitor” progress dialogs can overlap when many downloads are requested at the same time.
- Downloading the same video more than once at a time can cause errors.
- Some default settings are lost when switching languages.

## Future developments

MacYTDL is being actively developed. Functions which might be added in the future include:

- A “presets” facility.
- Option to choose items in playlists.
- Option to choose items in a batch.
- More credentials options: video password and *Adobe Pass*.
- Facility for controlling multiple *YT-DLP* config files.
- A file count in the download Monitor for playlist and multiple downloads.
- Scheduling downloads for a set time.
- A new “cookies from browser” setting.
- Some kind of “archive” function to save a list of past downloads which would be easier to use than the collection of log files. Perhaps with a tool to search log files.
- Facility to manage the “*.netrc*” file: add, delete, edit entries.
- A way to specify desired codecs.
- A setting to automatically make “*.part*” files playable – note that such files can be made playable by deleting the “*.part*” extension.
- In a future version, MacYTDL might be limited to users on macOS 10.14+. This is because *yt-dlp* is no longer updated for older versions of macOS and the *Alerter* tool is also limited to macOS 10.14+.
- The Monitor dialog could show the number of videos being downloaded for playlists, batches and multiple download situations. Note that the names of all videos are visible in the parallel Monitor dialog.
- It might be possible to revive the SBS OnDemand chooser to enable episodes of shows to be selected.
- Facility to choose to download particular chapters of a video when chapters are available.
- Capability to customise settings for individual items in a batch.

## Why use a video downloader ?

There is some controversy around the practice of downloading videos especially from YouTube and TV catch-up sites. Copyright owners are concerned people will copy and distribute videos and so reduce the revenue from DVD and online sales. It is a real fear as downloaded videos can easily end up listed on Bit Torrent web sites. That is why so many Bit Torrent sites have been blocked by court order in Australia and other countries. There is an active debate on the issue of such ‘piracy’ which would take too much space to cover here. But it’s fair to conclude no one has won the debate as yet. Even reputable commentators provide advice on [how to bypass blocks](#). Certainly, spreading copies of videos around the web via Bit Torrent is unethical. But, there are ethical reasons for downloading videos for personal viewing.

One reason for using a video downloader is particularly relevant to people in areas with poor internet access. When downloads are very slow or unreliable it is just not feasible to watch videos online using streaming services. With slow download speeds, it can take 2 hours or longer to watch a 1 hour video. If the video stream is interrupted, some sites require the viewer to start again. Downloading a video means it is possible to watch the full file without interruption.

Secondly, watching a downloaded video inside dedicated player software is much more flexible than watching a frame inside a web browser window. This is despite recent improvements in player tools in web browsers. Controlling a video on TV catch-up web sites is often quite horrible: cumbersome, flaky and slow, often with poor quality. Most have advertisements. Adverts are fine except that if there’s a streaming problem and viewing is interrupted, one usually has to wade through all the advertisements again to continue watching.

Just controlling web browser video players is hard. One has to muck with the mouse or keyboard to pause; fast forwarding and rewinding is unreliable; full screen display often isn’t; and play-back is sometimes interrupted or jerky even on a fast connection. Many people use a universal remote to control many functions on the Apple Mac they use to play videos eg. through a TV. A universal remote makes life much easier – it’s really hard to use a keyboard and mouse on your lap in front of the TV ! But, universal remotes (and especially the Apple Remote) usually can’t fully control a video playing in a web browser window.

In September 2020, on request of the Recording Industry Association of America (RIAA), the host GitHub disabled *youtube-dl*. In November 2020, *youtube-dl* was restored following a [letter](#) sent to GitHub by the Electronic Frontier Foundation. The letter explained why *youtube-dl* should remain available. It listed uses to which *youtube-dl* can be put:

“It is used by journalists and human rights organisations to save eyewitness videos, by educators to save videos for classroom use, by YouTubers to save backup copies of their own uploaded videos, and by users worldwide to watch videos on hardware that can’t run a standard web browser, or to watch videos in their full resolution over slow or unreliable Internet connections.”

Given those reasons, using a video downloader can be ethical. But, I think it’s best to adhere to a few more principles:

- Do not provide copies of downloaded videos to anyone else – especially do not provide copies online to other people by any means.
- Watch videos then delete them – the result then is similar to what would happen if recording a broadcast with a PVR for playback later.

- Look for other ways to access videos (eg. buy DVDs, borrow DVDs from the local library, record from free-to-air broadcasts, record or download from Pay TV providers who sell that product).
- If a broadcast is missed look for a repeat or “encore” broadcast before downloading.
- Send money to video authors, especially those on YouTube. Many authors make revenue from advertisements which are not played when a video is downloaded.

Disclaimer: Providing MacYTDL or this information is not to be taken to endorse any breaches of the law or of the terms and conditions imposed by owners of web sites and videos. You should check for terms and conditions on a web site before downloading.

## **MacYTDL components**

MacYTDL includes a number of AppleScript components.

### ***DialogToolkitMacYTDL.scptd***

This is an AppleScript library that must be installed for MacYTDL to work. It is stored in the MacYTDL bundle resources folder. The installer copies it from there to the “~/Library/Script Libraries” folder for each user who runs MacYTDL. It is automatically updated as needed.

It is a customised version of the *DialogToolkitPlus* script library created by [Shane Stanley, AppleScript Guru Extraordinaire](#).

### ***Myriad Tables Lib.scptd***

This AppleScript library is used to create the Formats List dialog. It is stored in the MacYTDL bundle resources folder. The installer copies it from there to the “~/Library/Script Libraries” folder for each user who runs MacYTDL. It was also created by [Shane Stanley](#).

### ***Main.scpt***

An AppleScript which provides the Main dialog and access to all MacYTDL functions. It is stored in the MacYTDL bundle.

### ***Monitor.scptd***

An AppleScript bundle which handles downloads and shows progress. Having a separate script enables MacYTDL to download a number of videos at the same time. It is stored in the MacYTDL bundle.

### ***adviser.scptd***

An AppleScript bundle which tracks when a download has finished. If the download was 100% complete it issues a notification accordingly and provides the option of playing the video. If not 100% complete or if an error is reported, the notification advises that there may have been a problem and provides options to open the related log file or the log folder. It is stored in the MacYTDL bundle.

### ***batch.scptd***

An AppleScript bundle containing AppleScript handlers which provide batch download functions.

### ***Utilities.scptd***

An AppleScript bundle containing a number of handlers which are called by the main script when required. Using a separate file for a number of handlers helps reduce crashes in my favourite AppleScript IDE. It is stored in the MacYTDL.

### ***Formats.scptd***

An AppleScript bundle used to create and show the “Choose formats” dialog. Is used if “Get formats list” is chosen in the “Settings” dialog. It is stored in the MacYTDL.

### ***Auto\_download.scpt***

An AppleScript bundle containing handlers which provide the automatic download capability.

### ***Send-URL-To-MacYTDL.workflow***

This is a macOS service created in Automator. It enables users to invoke MacYTDL from within other applications. It is stored in the MacYTDL bundle resources folder. When requested by the user, the installer copies it from there to the “~/Library/Services/” folder for each user who runs MacYTDL. When running, the usual rotating wagon wheel icon will show in the “Status” menu bar. Each user’s copy is automatically updated as needed if it is installed.

### ***Localisation files***

There are localisation files for German, Spanish, French, Italian and Portuguese. The localisations have been done to varying degrees, Spanish is the best. They are plain text files called “MacYTDL.strings” stored in separate folders named for each language. They are stored the MacYTDL bundle.

Lots of help is needed to improve localisation in MacYTDL.

## **External components**

MacYTDL uses seven tools to download and convert video files.

### ***YT-DLP***

*YT-DLP* is based on *youtube-dl*. The *youtube-dl* developers' description is best:

“youtube-dl is a command-line program to download videos from YouTube and a few [more sites](#). It requires the Python interpreter (2.6, 2.7, or 3.2+), and it is not platform specific. It should work on your Unix box, on Windows or on macOS.”

*YT-DLP* can download videos from many sites around the world including some not on their supported sites list. More detail on *YT-DLP* is available at <https://github.com/yt-dlp/yt-dlp>.

The unpackaged version of *YT-DLP* is used. A folder named “\_internal” is created to hold support files including Python, Cryptome, curl, web sockets, type guard and setup tools. The “\_internal” folder and *YT-DLP* binary are stored in the /usr/local/bin/ folder.

*YT-DLP* can be downloaded separately at: <https://github.com/yt-dlp/yt-dlp/releases>.

Installing *YT-DLP* takes around 175MB (which includes a Python runtime).

### ***Deno***

*Deno* is a Java runtime utility. It is recommended by *yt-dlp* for downloading videos from YouTube. Detail on *Deno* is available at <https://deno.com/>. It is stored in the /usr/local/bin/ folder.

Installing *Deno* takes around 105MB.

### ***Python scripting language***

Python is a script-based programming language that works on most operating systems. More detail is available here: <https://www.python.org>. Apple Macs running macOS 12.2.1 and earlier should have a workable copy of Python (usually version 2.7). However, if there are problems with Python, releases are available from: <https://www.python.org/downloads/mac-osx/>.

*YT-DLP* requires at least Python version 3.6. A Python runtime is included in the *YT-DLP* download used by MacYTDL. Thus, there is no need to install Python3 separately.

### ***FFmpeg***

*FFmpeg*, to quote from the developers, is:

“A complete, cross-platform solution to record, convert and stream audio and video.”

It is used by *YT-DLP* for a number of functions including to convert downloaded videos.

*FFmpeg* can be downloaded separately at: <https://ffmpeg.martin-riedl.de> [for macOS 12+] and <https://evermeet.cx/ffmpeg/> [for macOS 11 and earlier].

*FFmpeg* is stored in the “/usr/local/bin” folder.

Installing *FFmpeg* takes around 63MB.

## ***FFprobe***

*FFprobe* is a stream analyser. It is used by *YT-DLP* to collect details on the download video format etc.

*FFprobe* can be downloaded separately at: <https://ffmpeg.martin-riedl.de> [for macOS 12+] and <https://evermeet.cx/ffmpeg/> [for macOS 11 and earlier].

*FFprobe* is stored in the “/usr/local/bin” folder.

Installing *FFprobe* takes around 63MB.

## ***AtomicParsley***

*AtomicParsley* is a small utility that embeds thumbnail images in audio files. It is the default tool required by *youtube-dl* for that function. The *AtomicParsley* executable used by MacYTDL is version 20210715.151551.0, downloaded from the “[Wez/AtomicParsley](#)” repository on GitHub. I built a separate ARM version then combined that with the Intel version to produce a Universal 2 build. *AtomicParsley* thus runs natively on both Intel and ARM Macs.

That executable is stored in the MacYTDL bundle resources folder. When requested, the installer copies *AtomicParsley* to “/usr/local/bin”.

*Note:* *AtomicParsley* is not used by *YT-DLP*. It is being retained for those users who currently use *youtube-dl*.

Installing *AtomicParsley* takes around 730KB.

## ***Alerter***

*Alerter* is a utility which issues custom macOS notifications. It is used in MacYTDL to notify when components are being downloaded and when a download has finished.

Because the current version of *Alerter* does not run on versions of macOS prior to macOS 10.15 (Catalina), I built a separate copy in *Xcode* using a duplicate of the *Alerter* source code. The *Alerter* source code was for version 4 released on 27 December 2020. It has been tested to work on macOS 10.13 (High Sierra) and 10.15. The executable is stored within the MacYTDL bundle and is signed with the MacYTDL developer’s signature.

*Alerter* is built for Universal 2 and so runs natively on both Intel and ARM Macs. *Alerter* is stored within the MacYTDL bundle.

More detail on *Alerter* is available at: <https://github.com/vjeantet/alert>.

## ***ets***

*ets* is a utility which adds timestamps to the output of shell commands. It is used by MacYTDL to add timestamps to the log files, when requested by the user. I built a separate ARM64 version then combined that with the x86-64 version to produce a Universal 2 build. *ets* thus runs natively on both Intel and ARM Macs. *ets* is stored within the MacYTDL bundle.

More detail on *ets* is available at: <https://github.com/zmwangx/ets>.

## **Any comments, suggestions or better code ?**

Although MacYTDL has been in development for over 8 years, I still need all the help available. If you use MacYTDL and have any comments or suggestions please make contact.

The *AppleScript* code in MacYTDL is strictly an amateur effort. If you have *AppleScript* skills and would like to contribute please go ahead. It is best to use the facilities of GitHub to document proposed changes e.g. in pull requests. Note, however, that I normally only publish changes with each release and there will be changes that have not been published. So, please make contact before embarking on significant work.

MacYTDL has been developed with a commitment to some guiding principles. Those principles are set out here: <https://github.com/section83/MacYTDL/blob/master/Contributing.md>.

Send an email to [macytdl@gmail.com](mailto:macytdl@gmail.com) or post an issue on the MacYTDL GitHub page:

<https://github.com/section83/MacYTDL> [logon required to post]

## Software Licenses

The MacYTDL Software can be used for personal, non-commercial purposes and can be distributed, copied or uploaded without the author's permission. However, the license of the software must be mentioned when distributing the software for any other third party and a notification must be sent to the author ([macytdl@gmail.com](mailto:macytdl@gmail.com)) when any of the MacYTDL source code is changed or used in other software. The Software may not be used for direct or indirect commercial (includes non-profit, or revenue-generating business) purposes. You use it entirely at your own risk.

*AtomicParsley* is protected by the [GNU General Public License version 2.0 \(GPLv2\)](#).

*youtube-dl* is in the public domain, which means you can freely use it, modify it or redistribute it.

*YT-DLP* is in the public domain. It is covered by the [Unlicense License](#).

*Deno* is open source. It is covered by the [MIT License](#).

*Dialog Toolkit Plus* and *Myriad Tables Lib* are copyright ©2014-19, Shane Stanley. They may be freely distributed and copied. If you modify it, please rename the modified version to avoid clashes. You use it entirely at your own risk. Feedback is welcome.

*FFmpeg* is a trademark of [Fabrice Bellard](#), originator of the *FFmpeg* project. *FFmpeg* is licensed under the [GNU Lesser General Public License \(LGPL\) version 2.1](#) or later.

*Alerter* is protected by [the MIT License](#).

*ets* is protected by [the MIT License](#).

## Acknowledgements

Much is owed to Shane Stanley, for his many contributions solving problems with MacYTDL. Shane developed *Dialog Toolkit Plus* and *Myriad Tables Lib* which provide the main dialogs in MacYTDL.

Ideas for this GUI front-end came from many sources including:

- “Tombs” who was an active contributor to [Whirlpool](#) and author of the *URLDown Dropper* utility and various download scripts for Windows (discontinued).
- Adam Albrec ([game\\_creator@hotmail.com](mailto:game_creator@hotmail.com)), author of *PPC Media Centre*.
- Michael Page (<http://techion.com.au>), author of the *Video Hoarder* automator script.
- “kopurando” (<https://github.com/kopurando>), author of the *Virga* downloader.
- “xplorr” (<https://forum.videohelp.com/members/268051-xplorr>), author of *TVDownloader*.
- Anonymous (<https://cresstone.com/apps/youtubeDLFrontEnd/>), for *youtubeDLFrontEnd*.
- MacYTDL user Michel who maintains contact from release to release providing early warning of bugs and Spanish language help.
- Frank, Peter, Geoff, John, Santo, Trevor and Walter, fellow members of the [ACT Apple Users Group](#).
- All the MacYTDL users who've made contact: 1alessandro1, 11lucasarr11, Adam, adam01212, adenosslept, Adrenalyzer, Ashley, AirMarty, Alex, Alex Luis, alikaylan, alphabitnz, Andy, andyrb412, Anjum, Antaro, artcore-c, avcodex, barney1903, Başar, Bohdan, BigJoe309, bootyberry, bovirus, Brandon, CdrSpock, CharlesLai0307, Chris, clubizarre, ComfortableMilk4454, Dantha, darbid, defcon5at, Didier EuronymousDeadOhlin, Dorkington, dragonlord66666, EricTheDerek, Fred B, frissonlabs, froggyking3, Gábor Librecz, Grady, GrantGochnauer, gustavosaez, Hamza, heviiguy, hunterbr3193, Hussain, iluvcrap2000, inb4ohnoes, Infinivibex, ItsMorePaul, Ixne, Jack, JAKHIGDON, janvdvelde, JeanT, jeremydouglass, John @ TRIAL Media, John E, Joshua, L-Kiewa, kuglee, Labhansh-Sharma, Labhansh-Sharma, Larry, LeonardoMaracino, leon-chen-wen-jia, LordB54, lxne, marksinclair1, macmeister1967, macmeister1967, MalEbenSo, Marius, marksinclair1, martinbenny, martinsstuff, mcdiarimid1, meiwechner, michel-GH, Mihokas99, Mike, minkses, mmaslar, mmicha, MrJmpl3, Nellio, NorthEndVA, nottooloud, onaforeignshore, opartensky, palomnik, pedrocadiz13, Peter, QAQDE, Rick, Raymond-Adams, roelc500, roest01, Ruben, Scott, Stephenzero, SwineBurglar, tabascoman77, Ted, Tenz14, thejasonparker, tht7, tigrr, Tobias, Tom, TomasCarlson, tryitagain, upekshapriya, Vinsamlegast78, vorob1, watto23, williamcorney, Woolfy025, Yehushupat, zxzzz8.

## **Code signing and notarisation**

MacYTDL is code signed by me and notarised by Apple. These are separate functions which improve the security of macOS applications. They reduce the likelihood that an application has been interfered with after it was released by the developer.

Code signing is a process by which an application is “signed” with a certificate issued by Apple to the developer. macOS will not start an application that has been changed in any way after it is signed. Thus, there is some surety that the application has not been altered by another party eg. to add malicious code.

Notarisation is an additional process in which a copy of the application is examined by Apple to ensure it is signed correctly and contains no malicious code. Apple’s notarisation process is automated. It is not a formal review by an Apple developer. However, it does provide some comfort that Apple have checked the application and found nothing malicious.