

# 1. MCUtils

This package contains a suite of scripts for acquiring and manipulating MC metadata, and for performing various actions. The available scripts are listed below. The scripts are written in Perl so that they can be used on Windows, Mac OS X, or Linux.

Script	Description
amg.pl	Retrieve metadata from AMG to tag your Media Center tracks
discogs.pl	Retrieve metadata from Discogs to tag your Media Center tracks
pscriptor.pl	Extensible <i>scriptlet</i> -driven tool to manipulate, or do work based on, files and file metadata (fields)

## 1.1 Package Contents

This package contains the following:

- Several README files
- A Changes.txt change log
- The scripts amg.pl, discogs.pl, and pscriptor.pl
- A configuration file config.txt
- Several subdirectories containing Perl modules used by the scripts

## 1.2 Using the Package

Read and follow the instructions in this file **1\_README\_FIRST.pdf** and then in the files **2\_README\_INSTALL.pdf** and **3\_README\_USAGE.pdf**.

## 1.3 Script Descriptions

The scripts contained in the MCUtils package are more specifically described below.

---

## 1.4 Script: amg.pl and discogs.pl

The amg.pl and discogs.pl scripts pull metadata from AMG and Discogs, respectively, so that you may tag your Media Center (MC) files. Additionally, these scripts can be used to create MC entries for situations where physical media does not exist (e.g. Vinyl, desired CDs, etc.). This is accomplished by creating small *dummy* files that are then imported and tagged based on the desired AMG or Discogs metadata.

The scripts are used essentially as follows:

1. You identify and supply an album's identifier (AMG album ID / Discogs release ID)
2. You select the album's tracks in an MC view
3. You run the script

The scripts will then immediately update MC fields via MCWS<sup>1</sup>. Alternately, the scripts can be run without MCWS, and can produce an MPL file used to update the MC fields when imported into MC via the command *File > Import Playlist*.

---

<sup>1</sup> Requires MC version 19.0.54+, Media Server needs to be running with authentication enabled

## 1.5 Available Metadata

The per-album and per-track tables below show the metadata that can be retrieved (when available).

	amg.pl	discogs.pl
Per-Album	Album ID and URL	Release ID and URL
	Album	Album
	Artist and Album Artist	Album Artist
	Genre	Genre
	Recording Date, Release Date	Recording Date, Release Date
	Ratings (AMG rating, raw and mapped)	
	Album Pick	
	Album Review	
	Awards (Billboard Singles, Billboard Albums, Grammy Albums)	
	Releases	
	Album Credits (full and name-only)	Album Credits (full and name-only)
	Styles, Moods, Themes	Styles
	Similar Albums	
		Country
		Publisher, Catalog # (combined and separate)
		Media Format (Vinyl, CD)
		Barcode & Other Identifiers
	Notes	
Disc Number, Total Tracks	Disc Number, Total Tracks	

	amg.pl	discogs.pl
Per-Track	Track ID and URL	
	Track Artist ID and URL	
	Track Composer ID and URL	
	Track Number	Track Number
	Track Name	Track Name, Track Heading
	Track Artist	Track Artist
	Track Composer, Track Featuring	Track Credits
	Track Pick	
	Track Review	
	Track Lyrics	
		Track Playback Range, Track Duration

---

---

## 1.6 Script: pscriptor.pl

The pscriptor utility is an extensible *scriptlet*-driven tool used to manipulate, or do work based on, file metadata (fields). In its simplest form, you can think of pscriptor as being like Media Center's expression language on steroids. Furthermore, it also allows you to perform additional actions based on values collected from a set of files, making its utility virtually limitless.

Pscriptor works on the selected set of files in Media Center, and when run with a specific scriptlet (or command line Perl script), will cause various actions to occur, ranging from simply updating one or more fields, mirroring a set of files and creating the accompanying playlist, or performing acoustic fingerprint analysis and metadata lookup. The action performed depends on the chosen pscriptor scriptlet, or the supplied Perl script code. Many sample scriptlets are available. To help further clarify what pscriptor can do, the currently included scriptlets are summarized as follows:

- Sample scriptlets that demonstrate how to code a scriptlet (simple and complex cases)
- Get a music file's embedded cover art dimensions
- Convert names in the form of Last, First into First Last
- Uppercase the first letter of a name, except for those sub-names defined in an exception list
- Swap values of the form role:Artist into Artist:role
- Report on fields with case differences
- Create m3u playlists from files in a view
- Randomly fills Playback Range with a given range for true random track sampling
- Acoustic fingerprint audio analysis for track detection and musicality information
- Detect and retrieve dates from arbitrary fields
- Copy the topmost field from an arbitrary grouping of files, and push it downward in the group
- Copy and merge all the values in a field, removing duplicates
- Provide a sequence of numbers
- Make a random playlist of N files and start playback to a specified zone
- Obtain a list of each file's full playlist paths
- Test / verify a media file
- Copy files with flexible folder and file names (including sequence #'s) and create M3U8 playlists
- Populates MC fields from values taken from a CSV file, using a key value

These are just some examples of the types of things that can be done with pscriptor. More scriptlets will be added as needed, the idea being that the body of scriptlets will increase over time. If you know Perl, you're free to experiment, create and share your own.

The scripts are used essentially as follows:

1. You select the desired files in an MC view.
2. You run the script with the desired scriptlet.

The script will then perform the scriptlet-specific actions which may result in the immediate updates of specified MC fields via MCWS<sup>2</sup>. When relevant, the scripts can produce an MPL file used to update the MC fields when imported into MC via the command *File > Import Playlist*.

---

<sup>2</sup> Requires MC version 19.0.54+, Media Server needs to be running with authentication enabled

## 2. MCUtils Installation and Setup

This package requires a few steps to prepare MCUtils for usage. Follow the steps below.

### 1. Unzip the MCUtils package

Place the unzipped MCUtils folder onto your Desktop. Once you become familiar with how to run the scripts provided in the MCUtils suite, you can move the folder to another location.

### 2. Create custom configuration file(s)

The stock configuration file is **config.txt**. You may edit it directly, but should not, because it will be replaced the next time you update MCUtils. To prevent this, minimally, you should create a common configuration file in which to store the various MCWS settings.

Duplicate the file **config.txt**, and name the copy **config-custom.txt**. Using a plain text editor such as NotePad or TextEdit.app, open **config-custom.txt**. Remove or comment out the last line of the file:

```
include: config-custom.txt
```

otherwise you'll get errors about recursive include files.

Next read the comments under the MCWS section, and configure the `MCWS_server`, `MCWS_username` and `MCWS_password` values. Note: You are not required to use MCWS, however, the steps below will differ, and are not yet fully documented here.

### 3. Configure Field mappings, and Options

Review the various Fields sections in your **config-custom.txt** configuration file, and read the comments (comments are lines that begin with a # character). Some of the fields map into stock MC fields, and for these, there is nothing you need to do.

For every field that does not have an MC counterpart, you will need to create a custom field in MC if you want it filled. You can use the default MC field name shown in the config file, or you may change the name in the config file and assign it to the same field name in MC.

### 4. Configure Options

The Options section in the config file allows you to configure how the program should behave each run.

The utilities that parse web pages (currently only `amg.pl`) will download *many* web pages by default, since the desired metadata is spread across many different pages. Each additional page pull increases the amount of time it takes to process your command. It is highly recommended that you configure the **Options** section to *exclude* all fields you do not want to use, or to *specifically include* only those fields you want. See the comments in the Options section for how this is accomplished.

### 5. Installation

**Windows:** Download and install the Recommended version of [Strawberry Perl](#) for your Windows version. You may also choose to use the Portable version instead of the Recommended version that is uses the MSI installer. Instructions below will differ only slightly.

**OS X:** For OS X Installation: This process is not fully documented yet. See the file README-install-OSX.txt for some old notes.

## 6. Install the Requirement Modules

**Windows:** After Strawberry Perl is installed, open a Windows command shell by going to Start, entering

```
cmd
```

into the Search box, and hit Enter. Note: If you installed the portable version of Strawberry Perl, instead of manually starting the Windows command shell `cmd.exe`, just launch the batch script named **portableshell.bat** - it is inside the folder where you unzipped the Strawberry Perl archive.

Note: If you were using ActivePerl previously, you should remove it if you no longer need it. Also, once removed, be sure to remove the path components:

```
C:\Perl64\site\bin;C:\Perl64\bin;
```

from your **PATH** environment variable. The modules required with Strawberry Perl will fail to build if you have the ActivePerl components in your PATH and the PATH components mentioned above precede the Strawberry Perl path components shown below:

```
C:\Strawberry\c\bin;C:\Strawberry\perl\site\bin;C:\Strawberry\perl\bin
```

There are several additional modules required to use MCUtils. One at a time, type and enter each of the commands below exactly as shown into the command shell, and press Enter. Wait until each module is installed before attempting to install the next module:

```
cpan install Date::Calc
cpan install Win32::Clipboard
cpan install XML::LibXML
cpan install XML::XPath
cpan install Roman
```

Note: some pscripator scriptlets will require additional modules. See the comments at the top of a scriptlet prior to first use.

## 3. MCUtils Useage Notes - Table of Contents

3. Table of Contents	8
3.1 Using MCUtils	9
3.2 Getting Help	9
3.3 Saving Your Data	9
3.4 Command Line Usage Tips	10
3.5 Usage Instructions for the amg.pl and discogs.pl scripts	11
3.6 Obtaining the RID	11
3.7 Create Links	12
3.8 Using amg.pl and discogs.pl	12
3.9 Creating Dummy Files	13
3.10 Usage Instructions for the pscriptor.pl script	14
3.11 Using pscriptor.pl	14
3.12 Using Inline Perl with pscriptor.pl	15

## 3.1 Using MCUtils

The process for using utilities in the MCUtils suite is essentially the same for each utility. The basic steps are:

1. In a command shell, change directories to the MCUtils directory.
2. Select your files in Media Center.
3. Run a specific utility to update, or work on/with, the selected files.

The steps below are common to each utility in the MCUtils suite. Specific steps and instructions will follow. After reviewing and performing the common steps describe here, read the utility-specific instructions.

### 1. Open the Command Shell Window and Set the Directory

Open your command shell window, and change directories into the MCUtils folder. Use the command appropriate to your OS:

```
cd %USERPROFILE%\Desktop\MCUtils      # for Windows
cd ~/Desktop/MCUtils                  # for OS X
```

### 2. Select the Files in Media Center

In Media Center, go to any view and show the file list. Select the set of files you want to work on. If this is your first time using the utility, you might select just a couple of files for now, perhaps all the tracks for a single album.

Note: If you are not using MCWS, be sure the Filename field is shown as a file list column, and now use MC's *Edit > Copy* to copy your file metadata onto the clipboard. Note: MC 20 / Mac has a bug, and does not copy the proper clipboard data, so MCWS is required

### 3. Run the Script

To run a script, use the following command format:

```
perl SCRIPT.pl OPTIONS
```

where *SCRIPT.pl* is the name of the script you want to use, and *OPTIONS* are specific options to a script.

When the command run is successful, your track data will be updated, or other script-specific actions will be performed. When not using MCWS, to update metadata, use MC's *File > Import Playlist* and select the newly created **import.mpl** file to update track metadata.

## 3.2 Getting Help

An option common to all scripts is the `-help` option. It can be used to see a list of command line options supported by a script. Examples:

```
perl amg.pl --help
perl discogs.pl --help
perl pscriptor.pl --help
```

## 3.3 Saving Your Data

Because there is no way to Undo an action performed by the scripts, you might want to first create a Library Backup in case something goes wrong, or you change your mind. It might also be sufficient to create an MPL of the selected files. Use MC's *File > Export Playlist*, select the *Format MPL Playlist* setting and choose the setting *Output Range: Selection*. You may then use this MPL file (via *File > Import Playlist*) to restore values if necessary.

## 3.4 Command Line Usage Tips

Here are some command line usage tips that can make entering command lines easier.

### 1. File- and Folder-name Autocompletion

You can hit Tab after some file or folder name has been partially typed, and the command shell will autocomplete or cycle through possible completions for you.

### 2. Quote Shell Special Characters

Use double quotes to protect any white space and shell special characters in the names of files and folders, or within command line arguments. Single quotes are acceptable on OS X.

### 3. Recall the Previous Command Line

- Use the Up Arrow key to recall the previous command line. You can now edit the command to execute the new command when you hit Enter. Down Arrow will recall the next command from the command history.

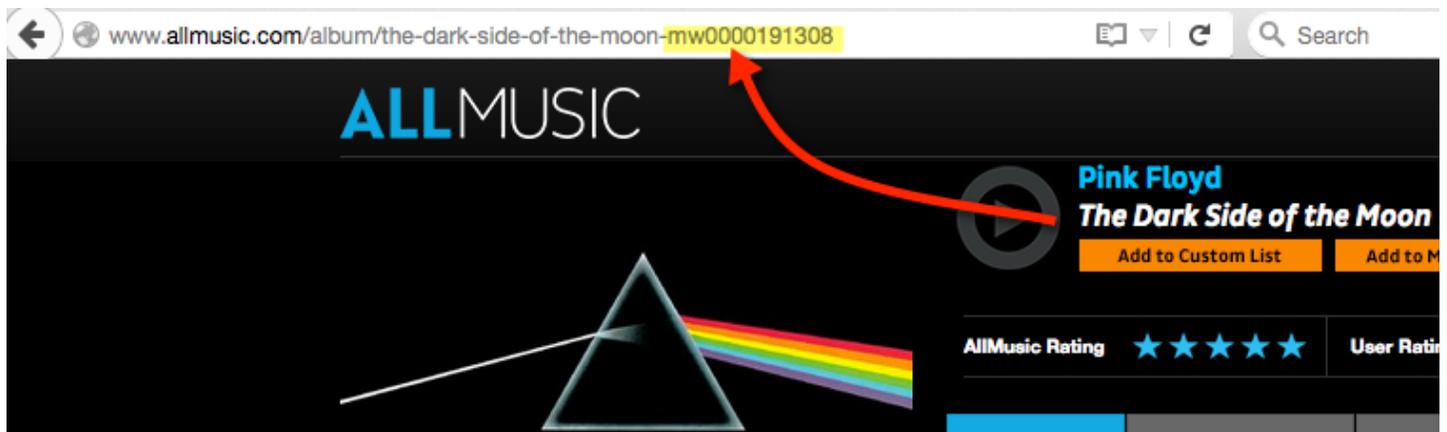
## 3.5 Usage Instructions for the amg.pl and discogs.pl scripts

The amg.pl and discogs.pl scripts will populate Media Center (MC) metadata with values pulled from AMG or Discogs. They work based on the selected set of files in the file list in an MC view, and it uses a unique AMG Album ID or Discogs Release ID (henceforth called the **RID**) in order to lookup the metadata. The scripts will retrieve a previously-stored RID from an MC field, or will use a value supplied on the command line using the `--rid` option.

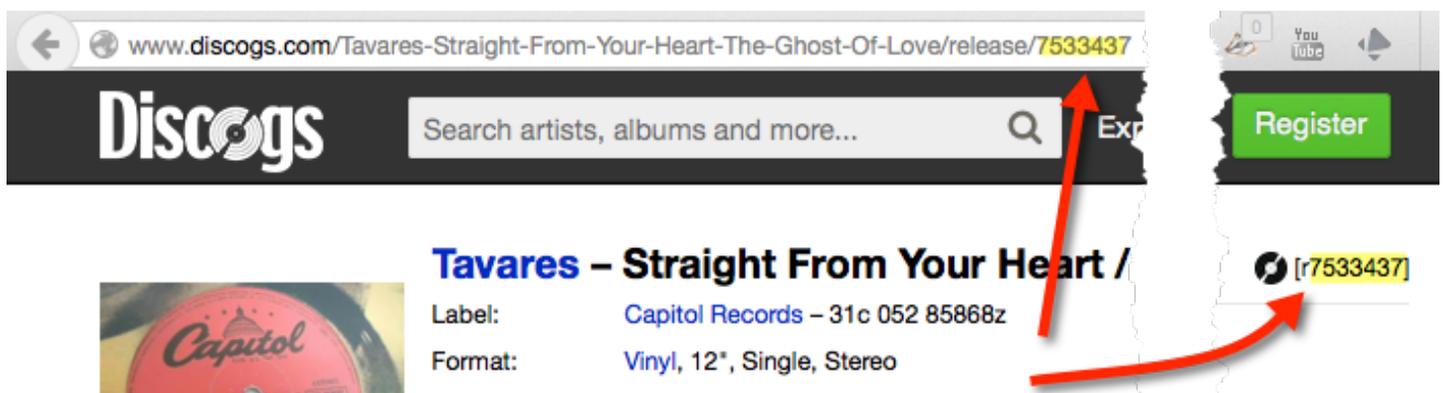
Key	AMG	Discogs
rid	AMG Album ID	Discogs Release ID

## 3.6 Obtaining the RID

**AMG RID:** To find the RID value for AMG, go to [www.allmusic.com](http://www.allmusic.com), and search for an Artist. Navigate into the artists discography listing, and select an album. Once you are on the album's page, copy the last part of the browser's URL, which will look something like `mw0002630894` or `mr0004109181`.



**Discogs RID:** To find the RID value for Discogs, go to [www.discogs.com](http://www.discogs.com) and search for an Artist. Navigate into the artists discography listing, and select an album. There will be one or more album releases; select your release (match your catalog #, or label, or use some generic entry if you don't need that level of detail - just be sure the release has the metadata you want). Once you are on the release page for the album, you'll see the Discogs Release ID as the last component of the URL, and also in the upper right corner of the page with a little record album icon (but you only want the numeric part). You'll want to copy the Release ID for later use.



---

## 3.7 Create Links

To facilitate searching for the RID, or later jumping to the specific page, you can create Links in MC, which when clicked, will open a webpage to the exact album, artist, etc.

---

## 3.8 Using amg.pl and discogs.pl

### 1. Obtaining and Using the RID

Probably the easiest way to initially populate metadata is to supply an RID on the command line. This value may then be stored in an MC field for later use. Read the config.txt file, and determine how you want to map the various IDs in Media Center. The default names for the IDs are listed below, and you can rename these via config file to match your MC field name(s).

Alternatively, the script will read the RID per-track from an MC field.

Note: If you are not using MCWS, then your MC field that contains the RID field must be in the view. Only files with an RID will be processed. Use the `--rid` option on the command line when a file does not have an RID value.

### 2. Building the Command Line

Assuming you are specifying the RID via command line, your command line will take one of the following forms, depending upon whether you want to run amg.pl or discogs.pl:

```
perl amg.pl --rid RID -v -t
```

```
perl discogs.pl --rid RID -v -t
```

replacing *RID* with the album's RID value. The `-v` option (a shorter form of the `--verbose` option) shows verbose output, and the `-t` option (short for `-testonly`) causes the script to only print the results, or print what would happen but not perform any actual action. This is useful to just testing what *would* happen). Add the command in the form above to the command line, but do not press *Enter* yet.

Note: If you do not want to use the MCWS service, include both the `--nomcws` and `--mpl` options.

### 3. Running the Command

If you are not using the MCWS service, now is the time to place the metadata for the selected tracks onto the clipboard by using MC's *Edit > Copy* command.

After you have entered your desired command, press *Enter* in the command shell to execute it. If the command worked correctly, you should see verbose command output that indicates the new tag values for your tracks as pulled from the source site. Since the `-t` test-only option was included, no values are yet written to MC. If the command did not work correctly, review any error messages to resolve the issue(s).

If the values appear to be correct, repeat the last command, this time without the `-t` option (and perhaps even without the `-v` verbose option). Hint: use the up-arrow to recall the last command, where you can quickly edit it.

After you've executed the command, the file data will be updated in MC (you may have to click in the MC window to cause it to refresh to view). Now is a good time to examine the new values in the view.

Note: If you are not using MCWS, your new metadata will be contained in the **import.mpl** file located on your Desktop. Use MC's *File > Import Playlist* command, selecting **import.mpl** to update the metadata.

---

## 3.9 Creating Dummy Files

These utilities can help you add virtual albums to MC from AMG or Discogs entries. They accomplish this by creating small *dummy* files that can be imported into MC along with the album and track metadata. . These dummy files are two-second, silent mp3 files, so they consume very little space. The process for creating and importing the dummy files is very simple.

Construct the command line as above, but this time include the options `--create` and `--basedir` (which indicates where the dummy files should be created). For example, to use `amg.pl` to create the dummy files, use the command form:

```
perl amg.pl -rid RID --create --basedir PATH -v -t
```

replacing *PATH* with a double-quoted path to a new folder where the dummy files should be created. Such a path might be `"C:\Dummy Files"`. With the `-t` option, the files are not actually created.

If the output from the script looks correct, you can remove the `-t` option. You should now see output that indicates the dummy files were created, and that they were created in a sub-folder named *RID*, under the base folder `"C:\Dummy Files"`. Finally, the command created a file named **import.mpl** on your Desktop. Import the newly created dummy files in MC by using *File > Import Playlist*, selecting the file **import.mpl**. These files are now imported into the MC library with the tags from AMG.

You can create and import more dummy files with different RIDs, and for faster processing, you can supply a comma-separated list of RIDs to the `--rid` option. Example using `discogs.pl`:

```
perl discogs.pl --create --basedir "C:\Dummy Files" --rid RID1,RID2,RID3 -v
```

## 3.10 Usage Instructions for the pscriptor.pl script

The pscriptor script will

XXX

---

## 3.11 Using pscriptor.pl

### 1. Initial Setup

By default, pscriptor works via MCWS to obtain values from your selected files.

When not using MCWS, MC fields you want to work with *must* be available as columns in the file list of a view. The Filename field must also be shown.

Some scriptlets will write to standard or custom MC fields. Review the scriptlet-specific instructions at the top of the scriptlet file in the Scriptlets directory, and then see the configuration section for the scriptlet in the **config.txt** file. It is beyond the scope of this overview document to explain script-specific details.

### 2. Building the Command Line

The command line for pscriptor will vary depending upon which scriptlet you want to use. In general, the pscriptor command line will take the form:

```
perl pscriptor.pl -E SCRIPTLET -f FIELD -v
```

and you replace *SCRIPTLET* with the scriptlet's name, and *FIELD* with the name of the MC field you want to affect.

Let's use a specific example for illustrative purposes, and assume you want to use the **NameCase** scriptlet to correct name-casing of the **Artist** field. For this example, the test-only command line would be:

```
perl pscriptor.pl -E NameCase -f Artist -v -t
```

The `-v` option (a shorter form of the `--verbose` option) shows verbose output, and the `-t` option (short for `-testonly`) causes the script to only print the results, or print what would happen but not perform any actual action. This is useful to just testing what *would* happen). Add the command in the form above to the command line, but do not press *Enter* yet.

Note: If you do not want to use the MCWS service, include both the `--nomcws` and `--mpl` options. Some scriptlets however cannot work without MCWS.

Note: Both *SCRIPTLET* and *FIELD* are case-insensitive.

Note: Don't forget to quote the *FIELD* name if it contains spaces or other shell-special characters.

Note: You may specify multiple `-f FIELD` pairs to affect more than one field, or supply a comma-separated list of *FIELD* names.

### 3. Running the Command

If you are not using the MCWS service, now is the time to place the metadata for the selected tracks onto the clipboard by using MC's *Edit > Copy* command.

After you have entered your desired command, press *Enter* in the command shell to execute it. If the command worked correctly, you should see verbose command output that shows your old and new field values output on the command line. Since we specified the `-t` test-only option, no values are written to MC yet. If the command did not work correctly, review any error messages to resolve the issue(s).

Examine the verbose output to check if the values appear to be correct, repeat the last command, this time without the `-t` option (and perhaps even without the `-v` verbose option). Hint: use the up-arrow to recall the last command, where you can quickly edit it.

After you've executed the command, the file data will be updated in MC (you may have to click in the MC window to cause it to refresh to view). Now is a good time to examine the new values in the view, or investigate the external changes made, depending upon the scriptlet's job.

Note: Some scriptlets do their work via side-effects - that is, they don't operate on a given field, but may do something like create a playlist file. Review the scriptlet-specific instructions at the top of the scriptlet file in the Scriptlets directory.

Note: If you are not using MCWS, your new metadata will be contained in the **import.mpl** file located on your Desktop. Use MC's *File > Import Playlist* command, selecting **import.mpl** to update the metadata.

---

## 3.12 Using Inline Perl with `pscriptor.pl`

The `pscriptor.pl` script was designed to read Perl script either via command line or via a scriptlet. You can use command-line Perl code by using the `-e` option (instead of the `-E` option). The argument to `-e` specifies the Perl code you want evaluate using the contents of the field. For example, the following command will remove all occurrences of bracketed values such as `[drums]` or `[piano]` from the Credits field:

```
perl pscriptor.pl -e "s/\[[^\]]+\]/g" -f Credits
```