

CATALOG: Digital Media Organization Language
Language Reference Manual
by Leonid Velikoselskiy (lvelik@gmail.com)

LANGUAGE

CATALOG is a shell-like language used to write programs to manipulate digital files such as images and music. It allows programmers to easily extract content-specific data from files and perform file manipulations easily.

CHARACTER SET

Any CATALOG program can be written using any eight-bit ASCII characters.

COMMENTS

Comments allowed in CATALOG are only single line comments starting with #.
everything to the left of # is compiled
everything to the right of # before the end of the line is ignored

CONSTANTS

Constants are strings and integers that represent values of file properties and patterns. Strings are any characters surrounded by double quotation marks and integers are any non-fractional decimals.

KEYWORDS

Following keywords are reserved and may not be used as variable names. They may however appear inside double quotation marks as strings or as parts of strings.

- cp
- mv
- del
- if
- foreach
- in
- inside
- sort
- by

VARIABLES

Variables must start with a letter or an underscore character and be followed by any number of digits, letters or underscore characters or by nothing. The following variable names are allowed:

```
_file  
file  
file2  
file_2
```

The following variable names are not allowed:

```
2file  
3.file  
4-file
```

CATALOG is typeless. Variables can be initialized and set to file handlers in the following way:

```
song = C:\\Music\\Pop\\FamousArtist\\NewAlbum\\NewHit.mp3
```

PATHS and FILE NAMES

Path to a directory or a file is described in the same way as the operating system on which the program runs describes it.

Names of directories are separated by either slash (/) or double backslash (\\) like this:

- C:\\Music\\Genre\\Artist
- c:/music/genre/artist
- /usr/local/music/artist

* (star) represents all files. File names are written as file name, followed by period, followed by file extension. Examples:

- favoriteSong.mp3
- house.jpeg

PROPERTIES

The purpose of CATALOG language is to work with two types of file: images and music files. All files in the system share some common properties. These general properties are:

- Name
- Extension
- Type
- Size
- DateCreated
- DateModified
- Attributes

For images, the most common format used in digital cameras is JPEG, therefore, programmers will be able to extract EXIF information from a file such as the following:

- CameraMake
- CameraModel
- DatePictureTaken
- ImageWidth
- ImageHeight

For music, the most common format for storing music used is MP3, therefore, programmers will be able to extract ID3 tag information from files such as the following:

- Artist
- Album
- Title
- Genre
- Year
- Track

So, for example, a program to move a file to a Pop directory if its genre is pop, will look like this:

```
$file = C:\\Music\\FamousArtist\\Song1.mp3
if $file.Genre = "Pop" mv $file C:\\Music\\Pop
```

There is also one dynamic property that is updated when the file is within a collection, it represents its index within that list:

```
CurrentIndex
```

So if a program is in foreach loop going through a collection of 10 files each file will be numbered 0 through 9 depending on their order.

PATTERNS

Property of each file can be formatted using a pattern inside parentheses in the following way:

PROPERTY(PATTERN)

There are two types of patterns:

date format pattern where in

- Y represents year digit
- M represents month digit
- D represents day digit

and substring pattern where

- positive number represents number of characters to return from the beginning of the file name
- negative number represents the amount of characters to return from the end of the file name

COMMANDS

- `cp PATH PATH` – copies file(s) from one path to another
- `mv PATH PATH` – copies file(s) from one path to another, optionally with new name(s), and deletes them from the original location.
- `del PATH` – deletes file(s) from a certain directory
- `if` – is not a command, but condition that checks for something being true and then executes a command

LOOPS

There are two ways to traverse through a file collection. VARIABLE is any user defined name and PATH is the physical path to the directory from which a list of files is being read. On each iteration of the loop VARIABLE is being set to the next file in the collection. Only the directory specified in PATH is traversed when `foreach ... in` keywords are used:

```
foreach VARIABLE in PATH {
    # do something
}
```

There is also an option to go through the directory specified in PATH and all of its subdirectories using `foreach ... inside` keywords.

```
foreach VARIABLE inside PATH {
    # do something
}
```

One other option is to apply a sort to the collection using one of the files' properties

```
foreach VARIABLE in sort PATH by property {
    # do something
}
```