NAME

DamScan – Daminion database analyzer

SYNOPSIS

```
Python DamScan.py [-f] [-i] [-g]
      [-t [{Event, Place, GPS, People, Keywords, Categories} [...]]]
      [-b [SEPARATOR [SEPARATOR ...]]]
      [-x EXFILE | -y ONLYFILE] [-a ACKFILE]
      [-l] [-c DBNAME] [-s SERVER] [-p PORT] [-u USER]
      [-o OUTFILE]
      [-v[v]] [-h] [--version]
```

DESCRIPTION

Daminion digital asset management (DAM) system is a great tool for assigning meta data (tags) to your digital assets and for sorting and searching the items. Daminion also allows you to link or group associated items together, but there are no built-in tools for checking the consistency of the meta data for the linked or grouped items. **DamScan.py** solves this problem and reports inconsistencies in metadata for Daminion server and standalone catalogs.

The **DamScan.py** tool analyzes a Daminion catalog for potential inconsistencies in tagging between linked or grouped media items. It checks the values of these tags: *Place, GPS, Event, People, Categories* and *Keywords*. It is also possible to verify, if the name parts of the files associated to linked or grouped media items are fully or partially identical. The command line options provide the capability to configure the process of the analysis as well as contents and format of its output.

For single value tags (*Place, GPS* and *Event*) and the program reports both file names and both tag values. For the multi value tags (*People, Categories* and *Keywords*) the program will report both filenames and which tag values **are missing** from the first file. If file name analysis is requested (option **-b**), the program compares if the specified part of the file names are identical and if not, report the full filenames of both media items.

The output is tab separated so it can be directly pasted or (when using option **-o**) imported into a spreadsheet for further formatting, sorting and analyzing.

By default the program analyzes the media items that are linked together and compares the tag values for each linked pair. Alternatively, when specifying the **-g** option, the behavior changes so that each image in a group (also called a stack) is compared with the top item.

Options:

-f, --fullpath   Display the full path and not only the file name

-i, --id   Display the Daminion Item Id after the filename

-g, --group   Compare tags between grouped images instead of linked images, if this option is not specified, links are used

-t, --tags [{Event,Place,GPS,People,Keywords,Categories} [{Event,Place,GPS, ... }] ...]]

Specify the tag categories that will be checked. If the **-t** option is not specified all of the tag categories listed above are checked. Allowed values for the list are *Event, Place, GPS, People, Keywords* and *Categories*. Multiple values are separated by spaces (' ').

Specifying **-t** without any tag category disables the tag value checking. This is only useful with the option **-b** to check only for consistent file names.

-b, --basename [*SEPARATOR* [*SEPARATOR ...*]]

Compare the base names (the name part excluding the extension, such as `.JPG`, `.CR2`, `.TIF` ...) of the file names associated to all linked or grouped (option **-g**) media items. If the names do not match, they are reported. If no *SEPARATOR* values are specified only the file extension is excluded from the comparison.

If one or more *SEPARATOR* values are specified (as single characters like '_' or as character strings like 'BW'), the file name is scanned from right to the first occurrence of the first *SEPARATOR* and only the characters on the left are considered for comparison or for subsequent search for the next *SEPARATOR*, if specified. If the *SEPARATOR* is not found from the name, the process continues with the next specified *SEPARATOR*, if any.

If you want to eliminate multiple occurrences of the same *SEPARATOR*, you need to specify the *SEPARATOR* multiple times (e.g. **-b** _ _ makes the file `IMG_1234_bw_lowres.JPG` to match `IMG_1234.JPG`).

If the name used for comparison would become shorter than 8 characters, the current and remaining *SEPARATOR*s will be ignored. This will avoid situation where specifying '_' as a *SEPARATOR* `IMG_0000.JPG` would match `IMG_9999.JPG`.

If the **-b** option is not specified, only tag values are checked.

Note

* the resulting base name for comparison can be different depending on the order of the *SEPARATOR*s.

* hyphen ('-') is allowed as a *SEPARATOR* alone. Multi character *SEPARATOR*s that start with a hyphen are not allowed, because they can be confused with the program options.

-x, --exclude *EXFILE*

> The option **-x** specifies a configuration file that contains tag values that will be excluded from comparison. The file format is similar to the standard .INI files. See details in the section CONFIGURATION PARAMETERS. Either **-x** or **-y** parameter can be specified, not both. If neither option is specified all tag values are checked.

-y, --only *ONLYFILE*

> The option **-y** specifies a configuration file that contains the only tag values that will be used for comparison. The file format is similar to the standard .INI files. See details in the section CONFIGURATION PARAMETERS. Either **-x** or **-y** parameter can be specified, not both. If neither option is specified all tag values are checked.
>
> Using the option **-y** with the same file as used with the option **-x**, makes it possible to verify what was or will be excluded.

-a, --acknowledged *ACKFILE*

> The option -a specifies a configuration file that contains acknowledged differences between linked or grouped media items. The differences listed in this file are excluded from the output. The output of the program, when used with the **-i** and **-o** option, can be used for a future run of the program with the -a option as an *ACKFILE*.

-l, --sqlite   Use a standalone (based on SQLite) catalog instead of server catalog (based on Postgresql)

-c, --catalog *CATALOG*

> The **-c** option specifies a Daminion catalog name. If not specified the default Daminion Server catalog (*NetCatalog*) is used. For standalone catalogs the full path and filename (including .dmc) must be specified.

-s, --server *SERVER*

> Postgres database server (**Not** the Daminion Server). If *SERVER* is not specified, localhost will be used. You can verify the *SERVER* and *PORT* settings in the Daminion Server Administration panel.

-p --port *PORT*

> Postgres database server port for the catalog. If not specified, the Daminion default *5432* will be used.

-u --user *USER*

> Postgres database user/password (**Not** Daminion catalog user). If not specified the installation default *postgres/postgres* will be used.

    -o --output *OUTFILE*

>Write the report to an *OUTFILE*. If **-o** is not specified the output will be printed on the screen. Verbose messages (**-v**) are never directed to *OUTFILE.*

-v, --verbose   Verbose output. Specifying the option **-v** displays a running counter, and the current Item Id and filename on the screen.

>If a second v is added to the option (**-vv**) also information of the linked or grouped pairs is printed. This output is always displayed on the screen (stdout) and not directed to the *OUTFILE* specified with the option **-o.**

-h, --help   Show help message and exit

--version   Display version information and exit.

DIAGNOSTICS

Errors and warnings are logged to the standard error stream and the diagnostic output to the standard output or the specified *OUTFILE*. If **-v** is not used, then no output means that no discrepancies were detected.

DamScan.py terminates with zero exit status if it was able to scan through the whole catalog.

Only the specified tag categories (Event, Place, GPS, People, Keywords, Categories) are checked, not all tag categories that are supported by Daminion. The other tag categories not in the list (e.g. Media Format, Rating, Project etc.) are ignored by **DamScan.py**.

When importing the output file into Excel, you have to select in import wizard at Step 1 **File origin:** *65001 : Unicode (UTF-8)*. This will import the accented and diacritic letters correctly.

Using **-g** option requires time on larger catalogs, because the scanning is n^2 dependent on the number of items in catalog. For example, for 150.000 items the analysis needs 1–2 hours, when the database and all executables where stored on an SSD and twice as much time on a hard disk.

ENVIRONMENT

Python

Install Python 3.x from http://www.python.org. After you have downloaded Python package right click the package and select "*Run as administrator*". In the installation dialog select Customized installation. In the customized configuration tick to include Python in the PATH and select installation for all users. Other options can be left to defaults.

To activate the PATH settings, Windows should be restarted before starting Python for the first time.

psycopg2

After installing Python start an elevated command window (*Run as Administrator*), because the Postgres support package will be installed in the Program Files directory. Enter commands

```
C:> python -m pip install -U pip setuptools
C:> python -m pip install psycopg2
```

CONFIGURATION PARAMETERS

EXFILE/ONLYFILE

The configuration parameters in the *EXFILE/ONLYFILE* follow the standard INI file structure. Each tag category can be specified in brackets ([category]) and the tag values below the section. If the tag value is hierarchical, the separator between hierarchy levels is '|'. If you specify only top items of a value hierarchy the filter applies to all child values.

Both the category names and tag values are case sensitive. You can specify also comment lines starting with '#' or ';'.

Below is an example configuration file.

```
[People]
# will be an exact match
Lintula|Juha

[Categories]
# will match to Image|B&W, Image|HDR, Image|Panorama, ...
Image

[Place]
# will match all cities and locations in Germany|Bavaria
Germany|Bavaria

[GPS]
# GPS coordinates not specified
0N 0E 0m
```

ACKFILE

The configuration parameters in the *ACKFILE* follow the structure of the program's output file. Each line contains an image pair, a tag category and the acknowledged differences. For the single value tags there is no need to specify any values. If the same image pair, tag category combination appears several times, all the listed tag values are taken into consideration.

The format for a single value tags (Name, Place, GPS or Event) is

```
file1 (id1)<TAB>sep<TAB>file2 (id2)<TAB>Name|Place|GPS|Event
   [<TAB>'value1'<TAB>sep<TAB>'value2']
```

and an example

```
160330_4440.JPG (7398)    <>    160330_4440.JPG (6670)    Place
```

sep is usually <>, < or >, but it could be anything. The vertical bar character ('/') separates different options and '*[text]*' optional values.

For the multivalue tags (People, Keywords, or Categories) the line is

```
file1 (id1)<TAB>sep<TAB>file2 (id2)
   <TAB>People|Keywords|Categories<TAB>'value1'[, 'value2'[, ...]]
```
and an example

```
160330_4456.JPG (6686)    <    160330_4456-001.JPG (7074)    Keywords
   'Palazzo|Collicola'
```
If you have hierarchical tags like 'Palazzo|Collicola', use the vertical bar character ('/') as a separator between different hierarchy levels.

If you run **DamScan** using option '**-i -o c:\DamScanResults.txt**', do not change the contents of file *DamScanResuls.txt* and run **DamScan** again with option '**-a c:\DamScanResults.txt**', it does not report any mismatches, unless new inconsistent pairs have been added to the Daminion catalog.

EXAMPLEs

Examples below assume that you have DamScan.py in your home directory (`C:\Users\user`) and your local catalog and the configuration files are in the `Pictures` sub-directory.

```
C:> python DamScan.py -s ServerPC -p 5433
```
Run the analysis of the NetCatalog server catalog, based on linked images. The Postgres database is set up in ServerPC at port #5433.

```
C:> python DamScan.py -v -g -l -c Pictures\DaminionCatalog.dmc -o
   Pictures\output.txt
```
Run the analysis of the local catalog `DaminionCatalog.dmc` in the `Pictures` directory, based on image groups. Print the results of the analysis in `Pictures\output.txt` and show basic progress information on the screen.

```
C:> python DamScan.py -t Place GPS -c NewCatalog -p 5433
```
Run the analysis only on Place and GPS tags (not on Event, People, Categories or Keywords) of the NewCatalog server catalog, based on linked images. The Postgres database is set up in localhost at port #5433.

```
C:> python DamScan.py -vv -x Pictures\ExcludeList.ini -c NewCatalog -o
   Pictures\output.txt
```
Run the analysis on all tag categories of the NewCatalog server catalog, but exclude the defined tag values from comparison. Print the results of the analysis in `Pictures\output.txt` and show detailed progress information on the screen.

```
C:> python DamScan.py -y Pictures\ExcludeList.ini -c NewCatalog
```
Check what was excluded from the report of the previous example.

```
C:> python DamScan.py -b -t -l -g -c Pictures\TestCatalog.dmc
```

Analyze only the filenames of the local `TestCatalog.dmc`, based on grouped images. Only the file extension is removed before the comparison. I.e. `IMG_1234.CR2` will be the same as `IMG_1234.JPG`, but different from `IMG_1234_lowers.JPG`.

```
C:> python DamScan.py -b _ BW -t -l -g -c Pictures\TestCatalog.dmc
```
Same as the previous example, but more logic for comparing the file names. First the file extension is removed. Then everything from the last '_' and after that everything from the last 'BW'. As a result all files `IMG_1234.JPG`, `IMG_1234_lowers.JPG`, `IMG_1234BW_lowers.JPG` and `IMG_1234BW.JPG` will match `IMG_1234.CR2`.

Examples of the use of the option **-b**.

File name	-b	Result	Comment
`IMG_1234.JPG`	-b _	`IMG_1234`	`IMG` is less than 8 characters
`IMG_1234_BW.JPG`	-b _	`IMG_1234`	
`IMG_1234_BW_lowres.JPG`	-b _ BW	`IMG_1234_`	_: `IMG_1234_BW` BW: `IMG_1234_`
`IMG_1234_BW_lowres.JPG`	-b BW _	`IMG_1234`	BW: `IMG_1234_` _: `IMG_1234`
`IMG_1234_BW_lowres.JPG`	-b _ _	`IMG_1234`	_: `IMG_1234_BW` _: `IMG_1234`

SEE ALSO

python, language description and syntax.

psycopg2, Python-PostgreSQL Database Adapter.

LICENSE

The program is licensed under GPL3.

AUTHOR(S)

Juha Lintula