NAME

DamCompare – Daminion database verification

SYNOPSIS

```
Python DamCompare.py [--ini INI_FILE] [-f] [-i]
     [-x|-y PATH [PATH ...]]
     [-l] [-c1 CATALOG1] [-c2 CATALOG2] [-s SERVER] [-p PORT] [-u USER]
     [-o OUTFILE]
     [-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 writes all tags (with few exceptions) into the media files, so that the metadata is also available outside Daminion. In Daminion there is no facility to verify if the metadata in the media items is the same as is in the Daminion database. **DamCompare.py** solves this problem by reporting inconsistencies in metadata for Daminion server and standalone catalogs. Inconsistencies can arise either by changing metadata in image file outside of Daminion, so the Daminion catalog is not aware of those changes or when changes in Daminion are not completely written into the metadata in the files.

The **DamCompare.py** tool analyzes a Daminion catalog for potential inconsistent tags in the Daminion database and the media items. The analysis is done by comparing two Daminion databases. The first one is the current active/production catalog and the second one is a newly to be created catalog representing current status of the image files. The current view is generated by creating an empty catalog and importing into it the same images that are in the first catalog. The import process will take time, because the speed can be as low as 15 items per minute depending on the size and utilization of the server.

The program checks the values of these tags: *Creation Date/Time, Place, GPS, Event, People, Categories, Keywords* and *Collections*. It also reports, if the media item is referring to a non-existing file by comparing if there is an entry for the same item in the second catalog. The command line options provide the capability to configure the process of the analysis as well as contents and format of its output.

For each media item the program reports its filename and all inconsistent tag categories. It's recommended to use the **-i** option, so that it is easy to find the mismatching items from the original and copy catalogs.

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

Options:

--ini *INI_FILE*

> A configuration file containing values for the command line options. If an option is specified both in the INI file and on the command line, the command line parameters have a precedence. See details in the section CONFIGURATION PARAMETERS.

-f, --fullpath Display the full path instead of only the file name

-i, --id Display the Daminion Item Id in parenthesis ( ) after the filename

-x, --excludepaths *PATH* (or discontinued **--exclude** *PATH*)

> Exclude those items which have matching beginning of the path. String `2017` matches both `2017\` and `2017-08-22\`, if you want to match a specific folder end the parameter with '`\`'. Use **-f** option to verify how the paths are stored in the Daminion database. Either **-x** or **-y** parameter can be specified, not both. If neither option is specified all paths are checked.

-y, --onlypaths *PATH* (or discontinued **--only** *PATH*)

> Compare only those items which have matching beginning of the path. String `2017` matches both `2017\` and `2017-08-22\`, if you want to match a specific folder end the parameter with '`\`'. Use **-f** option to verify how the paths are stored in the Daminion database. Either **-x** or **-y** parameter can be specified, not both. If neither option is specified all paths are checked.

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

-c1, --catalog1 *CATALOG1*

> The **-c1** option specifies the original Daminion catalog name. For standalone catalogs the full path and filename (including `.dmc`) must be specified.

-c2, --catalog2 *CATALOG2*

> The **-c2** option specifies the new created Daminion catalog against which the comparison is made. 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 displayed on the screen. Verbose messages (**-v**) are never directed to *OUTFILE*. To print the output on screen, when an output file specified in an INI file, a special value *<stdout>* can be used.

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

-h, --help Show help message and exit

--version **Only** displays program version information and exits.

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.

DamCompare.py terminates with zero exit status if it was able to scan through the whole catalog. If the *INI_FILE* contains unidentified keywords, the program issues a warning, but executes normally. If you are using the same *INI_FILE* for both **DamScan** and **DamCompare** you can ignore the warnings.

Only the specified tag categories (Creation Date/Time, Event, Place, GPS, People, Keywords, Categories, Collections) are checked, not all tag categories that are supported by Daminion. Other tag categories not in the list (e.g. Media Format, Rating, Project etc.) are ignored by **DamCompare.py**.

If the freshly created reference catalog contains only a small subset of the main catalog, it is recommended to specify the reference catalog as catalog1 and the main catalog as catalog2. Otherwise each item missing in the reference catalog will be reported as error. However, if you want to verify if items in the main catalog refer to nonexistent image files, then old one should be catalog1.

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 panel, select 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

INI_FILE

The file consists of two sections: Database and Session. Each line consists of one parameter, '=' sign and the parameter value. It's recommended to use full pathnames, so the INI file works independent of directory, where the **DamScan** program is run. The allowed parameters and there default values are:

[Database]

SQLite	Use a standalone (based on SQLite) catalog instead of server catalog, command line option **-l/--sqlite**. Allowed values: `True/False`. If defined `True` this cannot be reset on command line.
Catalog1	The Daminion catalog name (**-c1/--catalog1**). The `Catalog1` has to be specified either in the INI file or in the command line.
Catalog2	The Daminion catalog name (**-c2/--catalog2**). The `Catalog2` has to be specified either in the INI file or in the command line.
Server	Postgres database server (**-s/--server**). The default is `localhost`.
Port	Postgres database server port (**-p/--port**). The defauls is `5432`.
User	Postgres database user/password (**-u/--user**). The default is `postgres/postgres`.

[Session]

Fullpath	Display the full path and not only the file name (**-f/--fullpath**). Allowed values: `True/False`. The default value is `False`. If defined `True` this cannot be reset on command line.
ID	Display the Daminion Item Id after the filename (**-i/--id**). Allowed values: `True/False`. The default value is `False`. If defined `True` this cannot be reset on command line.
ExcludePaths	Specifies a list of directory paths separated by spaces (' ') that will be excluded from the comparison (**-x/--excludepaths**). If a path contains spaces, include the full path in double or single quotes ('"' or ''').

OnlyPaths   Specifies a list of directory paths separated by spaces (' ') that will be the only paths used for the comparison (**-y/--onlypaths**). If a path contains spaces, include the full path in double or single quotes ('"' or '''). If both Exclude and Only parameters are specified, a warning message is issued and the *ONLY PATH* is ignored.

Exclude   Same as **ExcludePaths.** Discontinued.

Only   Same as **OnlyPaths**. Discontinued.

Verbose   Verbose output. (**-v/--verbose**). Allowed values are 0 (default, no output), and 1. If this parameter has value 1, it cannot be reset to 0 on command line.

Output   Write the report to the specified file (**-o/--output**). If the parameter is not specified, the value is empty or the value is a special value <stdout> the output will be printed on the screen.

Below is an example INI file that corresponds to the defaults and has rest of the parameters commented out.

```
[Database]
SQLite=False
#Catalog1=NetCatalog1
#Catalog2=NetCatalog2
Server=localhost
Port=5432
User= postgres/postgres

[Session]
Fullpath=False
ID=False
# Exclude=path1 path2 path3 path4
# Only=path1 path2 path3 path4
Output=<stdout>
Verbose=0
```

EXAMPLEs

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

```
C:> python DamCompare.py -c1 NetCatalog -c2 NetCatalog1 -s ServerPC
   -p 5433
```
Run the analysis of the NetCatalog server catalog against NetCatalog1. The Postgres database is set up in ServerPC at port #5433.

```
C:> python DamCompare.py -v -l -c1 Pictures\DaminionCatalog.dmc
   -c2 Pictures\DaminionCatalogCopy.dmc -o Pictures\output.txt
```

Run the analysis of the local catalog `DaminionCatalog.dmc` in the `Pictures` directory and compare it to `DaminionCatalogCopy.dmc` in the same directory. Print the results of the analysis in `Pictures\output.txt` and show progress information on the screen.

```
C:> python DamCompare.py -v -l -c1 Pictures\DaminionCatalog.dmc
   -c2 Pictures\DaminionCatalogCopy.dmc -y .\Pictures\2017
   .\Pictures\2016-12\ -o Pictures\output.txt
```

As in the previous example, but run the analysis only for items in `.\Pictures` sub-folders `2017-01\`, `2017-02\`, ... and `2016-12\`.

```
C:> python DamCompare.py -v -l -c1 Pictures\DaminionCatalog.dmc
   -c2 Pictures\DaminionCatalogCopy.dmc -x .\Pictures\2017
   .\Pictures\2016-12\ -o Pictures\output.txt
```

As in the previous example, but exlude from the analysis items in `.\Pictures` sub-folders `2017-01\`, `2017-02\`, ... and `2016-12\`.

```
C:> python DamCompare.py --ini Pictures\DamCompare.ini -o <stdout>
```

Use the settings stored in an INI file `Pictures\DamCompare.ini`, but override to print the output on screen.

SEE ALSO

python, language description and syntax.

psycopg2, Python-PostgreSQL Database Adapter.

LICENSE

The program is licensed under GPL3.

AUTHOR(S)

Juha Lintula