

recreate_load_file

Rick Whitman
November 27, 1996

Usage

The tool is invoked by entering *recreate_load_file* on the command line followed by one argument which is a reference file name. For example:

```
recreate_load_file a3d1145dy.cy0
```

There are also two environment variables that must be defined in order for the tool to perform some database actions using OpenSTDB. These environment variables are:

- CDBS_SERVER - set to the database server name (in uppercase)
- CDBS_DB - set to the CDBS database name (in lowercase)

The output will be a single load file named the same as the input reference file name but with the extension “.lod”. For example:

```
a3d1145dy.lod
```

This file can be used simply as a reference or can be edited to update the comments. Then it can be used with the *comments_sql_gen* to update the comments in the database.

The tool will also output a line indicating the input file currently being processed and then the name of the output file produced to stdout.

```
Processing file A3D1145DY.CY0...  
Load file: a3d1145dy.lod has been generated
```

It will also write messages indicating errors, if they occur, to stdout. Upon detection of an error the task is terminated.

Algorithm/Module Description

The tool consists of a single file with several functions. It also calls functions from the *loadutils* library and the *OpenSTDB* library. The following steps are performed:

- The `GetInst` function is called to determine which instrument the reference file is associated with. This is accomplished by getting the last character in the file name before the “.” or “_”. This character is then used to query the `instrument_abbreviations` table in the database and return the instrument name. The file-level and row-level table names are determined from the instrument name.
- The `GetPutFileLevel` function is called to query the file-level table for the values associated with the reference file and output them to the load file.
- The `GetPutRowLevel` function is called to query the row-level table for the values associated with the reference file and output them to the load file.
 - This function first calls the `GetModes` function to determine which modes are used in this particular reference file type. It does this by querying the `cdbs_mode` database table with the given instrument and reference file type.
 - Then the row-level table for the instrument is queried for the keyword values associated with the reference file.
 - For each row-level row retrieved, the `GetChangeLevel` function is called to determine the `CHANGE_LEVEL` value for the load file. This is done by querying the row-level table for the equivalence class associated with the comparison file. Those values are then compared to the equivalence class values for the file being processed to determine a level of `SEVERE`, `MODERATE` or `TRIVIAL`.

Testing

The test script `recreate_load_file.sh` is available to test the `recreate_load_file` tool. The test script is invoked by entering its name. It is assumed that you are running the script from the `cdbs/test/script` directory and that your data is in the `cdbs/test/data` directory and that the `cdbs/test/data/tmp` directory exists. The script uses following files:

- `u-init-y.lod` - this file is an example of an initial unexpanded fos load file. It loads data for the file: `u-init-y.cy0`
- `u-next-y.lod` - this is an unexpanded fos load which uses `u-init-y` as its reference file. This will allow for testing the determination of the `CHANGE_LEVEL`.
- `e-init-y.lod` - this file is an example of an initial expanded fos load file. It loads data for the file: `e-init-y.cy0`

- *e-next-y lod* - this is an expanded fos load which uses *e-init-y* as its reference file. This will allow for testing the determination of the `CHANGE_LEVEL`.

Note that these files are not intended to be accurate examples of CDBS load files but are only intended to demonstrate the various program paths.

The following steps will take place during the test:

- *cdb_sql_gen* will be run on all of the files to generate SQL for loading the database. The SQL will then be executed to do the inserts.
- *recreate_load_file* will be run for each of the reference files to regenerate the load files in the `cdb/test/data/tmp` directory.
- The new load file in the `cdb/test/data/tmp` directory will be differenced with the original load files in the `cdb/test/data` directory. There should be no differences.