NAME

SQLite2DBF – convert an sqlite database-table to FoxBase (dBase)

SYNOPSIS

sqlite2dbf –s [SQLite–file] [options]

or

sqlite2dbf [Common options]

DESCRIPTION

You can use a variety of graphical user interfaces, notably those which support the SQL language, to create and maintain SQLite databases. The same is no longer true for dBase, a database format which is considered outdated, nowadays. However, the file–format is still in use in some contexts.

With sqlite2dbf you can convert one table at a time from an SQLite database into a dbf–file (in FoxBase format) and so benefit from the available GUI–interfaces, without risking incompatibilities, where a dBase–file is needed.

OPTIONS

–s, ––source [PATH]   SQLite–file to read.

–c, ––config [PATH]   Configuration file for this transformation

–n, ––name [TABLE]   The name of the table from the SQLite–database to convert

–t, ––target [PATH]   Path to the dBase–file to be written.

–l, ––list   Show the list of available tables and exit

–o, ––out [PATH]   Use the table–name as file–name for the DBF–result, store output in PATH

––time [list]   Fields (table–columns) which shall be handled as timestamp values.

––date [list]   Fields (table–columns) which shall be handled as date–time values.

Common Options

–d, ––debug   Show debug–messages

–h, ––help   Show this message

–v, ––version   Show version and program information

EXAMPLES

List available tables in a SQLite database:
    sqlite2dbf ––list –s database.sqlite

Transform a table from the database to dBase, the resulting file will be named after the table:
    sqlite2dbf –s database.sqlite ––name table

Transform a table from the database to dBase, write the result to the target–file:
    sqlite2dbf –s database.sqlite ––name table –t /directory/file.dbf

Transform a table from the database to dBase, put the result in a named directory:
    sqlite2dbf –s database.sqlite ––name table –o /directory

Transform a table from the database to dBase, handle the named fields as dates:
    sqlite2dbf –s database.sqlite ––name table ––date "expired last_accessed"

As before but be verbose:
    sqlite2dbf −s database.sqlite −−name table −−date "expired last_accessed" −d

Use a user−defined configuration from config.txt for this transformation:
    sqlite2dbf −c /home/user/sqlite2dbf_config.txt

As before but overwrite the path to the source−file:
    sqlite2dbf −c /home/user/sqlite2dbf_config.txt −−source base.sqlite

ERRORS AND WARNINGS

sqlite2dbf does not return error−codes but writes errors and warnings to STDOUT. This mainly concerns cases, where a data−type from the SQLite−database cannot be converted for use in the dBase−file, probably when date− and/or time−fields are listed on the command−line. Please contact the author, if these issues seriously obstruct your work with sqlite2dbf. The converter should in any way create a useable dBase−file.

SOURCE CODE AND DEVELOPMENT

sqlite2dbf is developed in Ruby and can be installed as a Ruby−Gem. As Ruby is an interpreter−language, the source−code of the installed version is always accessible. You can also decompress the gem−file to take a look at the code.

AUTHOR

Michael Uplawski <michael[dot]uplawski[at]uplawski[dot]eu>