

GULP - A simple journaling archiver.

GULP is a free and open source incremental, journaling command-line archiver for Windows and Linux. Journaling means that the archive is append-only. When you add files or directories to the archive, both the old and new versions are saved. You can recover old versions by specifying version number.

Build

Source code is written in pascal and it can be compiled with Lazarus/FreePascal.

Requirements

- FreePascal >= 3.0
- Lazarus >= 1.6

Instructions

If you have problems with permissions don't forget to prefix with `sudo`

- Clone the Gulp repository :

```
git clone https://github.com/melchiorrecaruso/gulp.git
cd gulp
```

- Checkout the latest Gulp release:

```
git fetch -p
git checkout $(git describe --tags `git rev-list --tags --max-count=1`)
```

- Build Gulp:

```
lazbuild gulp.lpi
```

This will create the gulp application at `build/${TargetCPU}-${TargetOS}/bin`.

- Install the `Gulp` to `/usr/local/bin` by executing:

```
sudo cp build/${TargetCPU}-${TargetOS}/bin/gulp /usr/local/bin/gulp
```

To use the newly installed Gulp, quit and restart all running Gulp instances.

Usage

The first argument to gulp should be a function; either one of the letters or one of the long function names. A function letter need be prefixed with "-", and can't be combined with other single letter options. A long function name must be prefixed with "--". Some options take a parameter; with the single-letter form these must be given as separate arguments. With the long form, they may be given by appending "=value to the option".

SYNOPSIS

```
gulp [-] s --synch | r --restore | p --purge | l --list |
      c --check | f --fix [options ...] [ files ...]
```

Main function mode

Synchronize archive (-s, --sync)

Append changes in files to archive, or create archive if it does not exist. "files" is a list of file and directory names

separated by spaces. If a name is a directory, then it recursively includes all files and subdirectories within. In Windows, files may contain wildcards "*" and "?" in the last component of the path (after the last slash). "*" matches any string and "?" matches any character.

A change is an addition, update, or deletion of any file or directory in "files" or any of its subdirectories to any depth. A file or directory is considered changed if its size, last-modified date, or Windows attributes or Unix/Linux permissions differ between the internal and external versions. File contents are not compared.

For each added or updated file or directory, the following information is saved in the archive: the contents, the file or directory name as it appears in "files" plus any trailing path, the last-modified date and the Unix/Linux permissions or Windows attributes. Other metadata such as owner name, group name, last access time, etc. are not saved. Symbolic links are saved. Hard links are followed as if they were ordinary files. Special file types such as devices, named pipes, and named sockets are not saved. If any file cannot be read (e.g. permission denied), exception is raised.

Note :

- Updates are transacted. If GULP is interrupted before completing the update, then the archive can be repair with fix function.

Example :

Create archive.gulp from directories foo and bar

```
gulp -s archive.gulp foo bar
```

Restore files and directories (-r, -restore)

Restore "files" (including the contents of directories), or extract the whole archive contents if "files" is omitted. The file names, last-modified date, and permissions or attributes are set as saved in the archive.

Note :

- If there are multiple versions of a file stored, then only the latest version is extracted.
- If a stored file has been marked as deleted, then it is deleted. Existing files are overwritten if they are considered changed.

Example :

Restore all files from archive.gulp into current directories

```
gulp -r archive.gulp
```

Purge archive (-p, -purge)

Purge archive, remove old files archived.

List archive contents (-l, -list)

List "files" within the archive, or list the entire archive contents if "files" is omitted. For each file or directory, show marker, last modified date, Windows attributes or Unix/Linux permissions, size and name.

Note :

- Attributes are listed as an octal number in Unix/Linux (as per chmod(1)) or with the letters D, A, S, H, R, I in Windows (as for the attrib command).

Example :

List all files in archive.gulp

```
gulp -l archive.gulp
```

Check archive integrity (-c, -check)

Check archive integrity by verifying that the data agrees with the stored SHA-1 hashes and sizes and that the data otherwise conforms to the gulp standard.

Fix damaged archive (-f, -fix)

Truncates any data added after last valid update.

Options

Exclude pattern (-e, -exclude=PATTERN)

Avoid operating on files whose names match filename pattern PATTERN. The option prevents any file or member whose name matches the shell wildcard (pattern) from being operated on. Multiple exclude options are supported.

Example :

List all files in archive.gulp but exclude .pas files

```
gulp -l archive.gulp --exclude='*.pas'
```

Force path (-forcepath)

It's recommended store files in archive with relative path names. This option forces GULP to operate on files with absolute path.

Include option (-i, -include=PATTERN)

Operating on files whose names match filename pattern PATTERN. Multiple include options are supported.

Example :

List *.pas files in archive.gulp

```
$ gulp -l archive.gulp --include=*.pas
```

Not delete files (-nodelete)

With synch command (-s), do not mark files in the archive as deleted when the corresponding external file does not exist. With restore command (-r), do not delete external files when the corresponding file in archive does not exist.

Note:

- This makes **GULP** consistent with the behavior of most non-journaling archivers.

-u, -until [version number]

Ignore any part of the archive updated after version number.

Example :

- restore content of archive until version 120 into current directories

```
gulp -r archivename.gulp -u 120
```

- show files added before version 21

```
gulp -l archive.gulp files --until 20
```

General file format version 0.3

A **GULP** archive is a sequence of timestamped updates, each listing the files and directories that have been deleted or added, since the previous transacted update, normally based on changes to the last-modified dates.

Archive content:

- Slice 1
- Slice 2
- ...
- Slice N

Slice content:

- Marker (20 bytes)
- Header 1
- Checksum 1 (20 bytes)
- Marker (20 bytes)
- Header 2
- Checksum 2 (20 bytes)
- ...
- Marker (20 bytes)
- Header N
- Checksum N (20 bytes)
- Stored data 1
- Stored data 2
- ...
- Stored data N

Header content:

- | | | | |
|---------------|----------------|------------|--------------------|
| - Flags | longword | (4 bytes) | |
| - Name | rawbytestring | | |
| - sTime (UTC) | double | (8 bytes) | |
| - mTime (UTC) | double | (8 bytes) | if bit 04 in Flags |
| - Attributes | longint | (4 bytes) | if bit 05 in Flags |
| - Mode | longint | (4 bytes) | if bit 06 in Flags |
| - Size | int64 | (8 bytes) | if bit 07 in Flags |
| - Link Name | rawbytestring | | if bit 08 in Flags |
| - User ID | longint | (4 bytes) | if bit 09 in Flags |
| - Group ID | longint | (4 bytes) | if bit 10 in Flags |
| - User Name | rawbytestring | | if bit 11 in Flags |
| - Group Name | rawbytestring | | if bit 12 in Flags |
| - Comment | rawbytestring | | if bit 13 in Flags |
| - Offset1 | int64 | | if bit 07 in Flags |
| - Offset2 | int64 | | if bit 07 in Flags |
| - Checksum | array of bytes | (20 bytes) | if bit 07 in Flags |

Note:

- bit 01 is reserved for mark header added to archive;
- bit 02 is reserved for mark header deleted from archive;
- bit 03 is reserved for mark last header in current slice.

Credits

GULP is copyright (c) 2014-2016, Melchiorre Caruso. It is licensed under The GNU General Public License v2. For

information on the license, see [GPLv2](#) . Program was written by Melchiorre Caruso, melchiorrecaruso (at) gmail (dot) com