
PyG3T Documentation

Release 0.5.0

Ask H. Larsen and Kenneth Nielsen

February 15, 2017

1	Getting Started	3
2	Examples	5
2.1	gtcat	5
2.2	podiff	5
2.3	gtcompare	5
2.4	Miscellaneous	6
3	API documentation	7
3.1	gtparse module	7
4	Glossary	9
5	Indices and tables	11

PyG3T (short for Python GetText Translation Toolkit) is a set of pure Python tools to work with gettext catalogs (.po-files) as a translator.

The toolkit consists of:

- **gtcat**: write a catalog in normalized format, or change encoding
- **gtcheckargs**: parse translations of command line options in a catalog, checking for errors (designed for GNU coreutils and similar)
- **gtcompare**: compare two catalogs qualitatively
- **gtgrep**: perform string searches within catalogs
- **gtmerge**: combine two or more catalogs in different ways
- **gtprevmsgdiff**: show a word-wise diff which compares old msgids in a catalog with current ones
- **gtwdiff**: show an ordinary podiff as a word-wise podiff
- **poabc**: check for common translation errors, such as missing punctuation
- **podiff**: generate diffs of po-files, such that each differing entry is printed completely
- **popatch**: apply a podiff to an old catalog to obtain the new catalog
- **gtxml**: check xml in translations

Contents:

Getting Started

TODO

Examples

gtcat

Write catalog to stdout with syntax coloring:

```
gtcat -c file.po
```

Change encoding of file to UTF-8:

```
gtcat --encode utf-8 file.po > file.utf8.po
```

podiff

The main application of podiff is to generate a diff where the translator has added and changed some strings or comments. Generate a diff of two catalogs:

```
podiff old.po new.po > differences.podiff
```

If the two catalogs have different English strings, the above will cause an error, because we expect only the translator to have been at work. To override this and produce a diff with all changes, including addition and removal of English strings, use:

```
podiff old.po new.po --full > differences.full.podiff
```

gtcompare

You get a file from Launchpad and another from debian for the same translation, but with different contents:

```
gtcompare file1.po file2.po
```

This will print something like:

```
Template of second file is more recent
Translations in first file were revised more recently

Total number of messages increased by 1849 from 193 to 2042.

49 msgids removed [u: 0, f: 0, t: 49].
1898 msgids added [u: 176, f: 307, t:1415].
```

```
144 msgids in common.
0 messages remain untranslated.
0 untranslated messages changed to fuzzy.
0 untranslated messages changed to translated.
0 fuzzy messages changed to untranslated.
0 messages remain fuzzy.
0 fuzzy messages changed to translated.
53 translated messages changed to untranslated.
87 translated messages changed to fuzzy.
4 messages remain translated.
```

Miscellaneous

Update an outdated podiff (program.podiff) to match the msgid of a newer version (master.po), and regenerate the diff:

```
popatch program.podiff --new > new.po
gtmerge new.po master.po > merged.po
podiff master.po merged.po > program.new.podiff
```

The diff will only include those messages that are still present in the new template. Often it might be a good idea to use msgmerge instead of gtmerge to preserve fuzzy messages.

This command does the same, but in one line:

```
popatch program.podiff --new | gtmerge - master.po | podiff master.po - > program.new.podiff
```

API documentation

Autogenerated API documentation for the pyg3t modules.

gtparse module

Summary

Catalog
Message
ObsoleteMessage
PoParser
parse

API

Glossary

chunk Chunk refers to the chunk of text in a .po file that corresponds to a *message*. The whole file consists of such chunks separated by a blank line. See the entry for *message* for an example of what a chunk looks like.

fuzzy Fuzzy is a flag that can be set programmatically when extracting strings for translation or manually by the translator. It is used to indicate that the translation may no longer be valid. Once the translator has reviewed (and possibly corrected) the string, the flag is removed.

See the [gettext reference documentation](#) for details.

gettext catalog A gettext catalog is a collection of gettext *message*s and a header with metadata. It is most often contained in a .po file. The header is the message with an empty *msgid* and is usually the first message of the file. In pyg3t a gettext catalog is represented by the `pyg3t.gtparse.Catalog` class.

See the [gettext reference documentation](#) for details.

message A message refers to all data and metadata pertaining to a single translatable string in a *gettext catalog*. The data consists of the *msgid* (original string) and the *msgstr* (translated string), both of which can optionally have plural forms. The metadata consists of references to source code lines for the msgids and optionally a comment, context and a previous version of the msgid. In pyg3t a message is represented by the classes `pyg3t.gtparse.Message` and `pyg3t.gtparse.ObsoleteMessage`.

A typical message could look like this (inspired by the Danish translation of nautilus) (FIXME better example):

```
# Ahh, nautilus can run programs
#: ../data/nautilus-autorun-software.desktop.in.in.h:1
msgid "Run Software"
msgstr "Kør programmer"
```

See the [gettext reference documentation](#) for details.

msg See *message*

msgid The original string (usually in english), which is to be translated.

See the [gettext reference documentation](#) for details.

msgstr The translated string.

See the [gettext reference documentation](#) for details.

Indices and tables

- `genindex`
- `modindex`
- `search`

C

chunk, 9

F

fuzzy, 9

G

gettext catalog, 9

M

message, 9

msg, 9

msgid, 9

msgstr, 9