



Rocky Enterprise Linux 9.2 Manual Pages on command 'xgettext.1'

\$ man xgettext.1

XGETTEXT(1) User Commands XGETTEXT(1)

NAME

xgettext - extract gettext strings from source

SYNOPSIS

xgettext [OPTION] [INPUTFILE]...

DESCRIPTION

Extract translatable strings from given input files.

Mandatory arguments to long options are mandatory for short options too. Similarly for optional arguments.

Input file location:

INPUTFILE ...

input files

-f, --files-from=FILE

get list of input files from FILE

-D, --directory=DIRECTORY

add DIRECTORY to list for input files search

If input file is -, standard input is read.

Output file location:

-d, --default-domain=NAME

use NAME.po for output (instead of messages.po)

-o, --output=FILE

write output to specified file

-p, --output-dir=DIR

output files will be placed in directory DIR

If output file is -, output is written to standard output.

Choice of input file language:

-L, --language=NAME

recognise the specified language (C, C++, ObjectiveC, PO, Shell,
Python, Lisp, EmacsLisp, librep, Scheme, Smalltalk, Java,
JavaProperties, C#, awk, YCP, Tcl, Perl, PHP, Ruby, GCC-source,
NXStringTable, RST, RSJ, Glade, Lua, JavaScript, Vala, Desktop)

-C, --c++

shorthand for --language=C++

By default the language is guessed depending on the input file name extension.

Input file interpretation:

--from-code=NAME

encoding of input files (except for Python, Tcl, Glade)

By default the input files are assumed to be in ASCII.

Operation mode:

-j, --join-existing

join messages with existing file

-x, --exclude-file=FILE.po

entries from FILE.po are not extracted

-cTAG, --add-comments=TAG

place comment blocks starting with TAG and preceding keyword
lines in output file

-c, --add-comments

place all comment blocks preceding keyword lines in output file

--check=NAME

perform syntax check on messages (ellipsis-unicode, space-ellip?

sis,

quote-unicode, bullet-unicode)

--sentence-end=TYPE

type describing the end of sentence (single-space, which is the

default,

or double-space)

Language specific options:

-a, --extract-all

extract all strings (only languages C, C++, ObjectiveC, Shell,

Python, Lisp, EmacsLisp, librep, Scheme, Java, C#, awk, Tcl,

Perl, PHP, GCC-source, Glade, Lua, JavaScript, Vala)

-kWORD, --keyword=WORD

look for WORD as an additional keyword

-k, --keyword

do not to use default keywords (only languages C, C++, Objec?

tiveC, Shell, Python, Lisp, EmacsLisp, librep, Scheme, Java, C#,

awk, Tcl, Perl, PHP, GCC-source, Glade, Lua, JavaScript, Vala,

Desktop)

--flag=WORD:ARG:FLAG

additional flag for strings inside the argument number ARG of

keyword WORD

(only languages C, C++, ObjectiveC, Shell,

Python, Lisp, EmacsLisp, librep, Scheme, Java, C#, awk, YCP,

Tcl, Perl, PHP, GCC-source, Lua, JavaScript, Vala)

-T, --trigraphs

understand ANSI C trigraphs for input (only languages C, C++,

ObjectiveC)

--its=FILE

apply ITS rules from FILE (only XML based languages)

--qt recognize Qt format strings (only language C++)

--kde recognize KDE 4 format strings (only language C++)

--boost

recognize Boost format strings (only language C++)

--debug

more detailed formatstring recognition result

Output details:

--color

use colors and other text attributes always

--color=WHEN

use colors and other text attributes if WHEN. WHEN may be 'always', 'never', 'auto', or 'html'.

--style=STYLEFILE

specify CSS style rule file for --color

-e, --no-escape

do not use C escapes in output (default)

-E, --escape

use C escapes in output, no extended chars

--force-po

write PO file even if empty

-i, --indent

write the .po file using indented style

--no-location

do not write '#: filename:line' lines

-n, --add-location

generate '#: filename:line' lines (default)

--strict

write out strict Uniform conforming .po file

--properties-output

write out a Java .properties file

--stringtable-output

write out a NeXTstep/GNUstep .strings file

--itstool

write out itstool comments

-w, --width=NUMBER

set output page width

--no-wrap

do not break long message lines, longer than the output page width, into several lines

-s, --sort-output

generate sorted output

-F, --sort-by-file

sort output by file location

--omit-header

don't write header with 'msgid ""' entry

--copyright-holder=STRING

set copyright holder in output

--foreign-user

omit FSF copyright in output for foreign user

--package-name=PACKAGE

set package name in output

--package-version=VERSION

set package version in output

--msgid-bugs-address=EMAIL@ADDRESS

set report address for msgid bugs

-m[STRING], --msgstr-prefix[=STRING]

use STRING or "" as prefix for msgstr values

-M[STRING], --msgstr-suffix[=STRING]

use STRING or "" as suffix for msgstr values

Informative output:

-h, --help

display this help and exit

-V, --version

output version information and exit

-v, --verbose

increase verbosity level

AUTHOR

Written by Ulrich Drepper.

REPORTING BUGS

Report bugs in the bug tracker at <https://savan?>

nah.gnu.org/projects/gettext> or by email to <bug-gettext@gnu.org>.

COPYRIGHT

Copyright ? 1995-2020 Free Software Foundation, Inc. License GPLv3+:

GNU GPL version 3 or later <<https://gnu.org/licenses/gpl.html>>

This is free software: you are free to change and redistribute it.

There is NO WARRANTY, to the extent permitted by law.

SEE ALSO

The full documentation for xgettext is maintained as a Texinfo manual.

If the info and xgettext programs are properly installed at your site,

the command

info xgettext

should give you access to the complete manual.

GNU gettext-tools 20200704

July 2020

XGETTEXT(1)