NAME

xgettext – extract gettext call strings from C programs

SYNOPSIS

xgettext [**–ns**] [**–a** [**–x** *exclude-file*]] [**–c** *comment-tag*] [**–d** *default-domain*] [**–j**]
[**–m** *prefix*] [**–M** *suffix*] [**–p** *pathname*] – | *filename* . . .
xgettext –h

AVAILABILITY

SUNWloc

DESCRIPTION

xgettext is used to automate the creation of portable message files (**.po**). A **.po** file contains copies of "C" strings that are found in ANSI C source code in *filename* or the standard input if '–' is specified on the command line. The **.po** file can be used as input to the **msgfmt**(1) utility, which produces a binary form of the message file that can be used by application during run-time.

xgettext writes *msgid* strings from **gettext**(3I) calls in *filename* to the default output file **messages.po**. The default output file name can be changed by –**d** option. *msgid* strings in **dgettext( )** calls are written to the output file *domainname***.po** where *domainname* is the first parameter to the **dgettext( )** call.

By default, **xgettext** creates a **.po** file in the current working directory, and each entry is in the same order the strings are extracted from *filenames*. When the –**p** option is specified, the **.po** file is created in the *pathname* directory. An existing **.po** file is overwritten.

Duplicate *msgid*s are written to the **.po** file as comment lines. When the –**s** option is specified, the **.po** is sorted by the *msgid* string, and all duplicated *msgid*s are removed. All *msgstr* directives in the **.po** file are empty unless the –**m** option is used.

OPTIONS

–**n** Add comment lines to the output file indicating file name and line number in the source file where each extracted string is encountered. These lines appear before each *msgid* in the following format:

#
 # File: *filename***, line:** *line-number*

–**s** Generate output sorted by *msgid*s with all duplicate *msgid*s removed.

–**a** Extract all strings, not just those found in **gettext**(3I), and **dgettext ( )** calls. Only one **.po** file is created.

–**c** *comment-tag* The comment block beginning with *comment-tag* as the first token of the comment block is added to the output **.po** file as # delimited comments. For multiple domains, **xgettext** directs comments and messages to the prevailing text domain.

–**d** *default-domain* Rename default output file from **messages.po** to *default-domain* **.po**.

–**j** Join messages with existing message files. If a **.po** file does not exist, it is created. If a **.po** file does exist, new messages are appended. Any duplicate **msgid**s are commented out in the resulting **.po** file. Domain directives in the existing **.po** file are ignored. Results not guaranteed if the existing message file has been edited.

–**m** *prefix*	Fill in the *msgstr* with *prefix*. This is useful for debugging purposes. To make *msgstr* identical to *msgid*, use an empty string (**""**) for *prefix*.
–**M** *suffix*	Fill in the *msgstr* with *suffix*. This is useful for debugging purposes.
–**p** *pathname*	Specify the directory where the output files will be placed. This option overrides the current working directory.
–**x** *exclude-file*	Specify a **.po** file that contains a list of *msgid*s that are not to be extracted from the input files. The format of *exclude-file* is identical to the **.po** file. However, only the *msgid* directive line in *exclude-file* is used. All other lines are simply ignored. The –**x** option can only be used with the –**a** option.
–**h**	Print a help message on the standard output.

SEE ALSO

msgfmt(1), **gettext**(3I)

NOTES

xgettext is not able to extract cast strings, for example ANSI C casts of literal strings to **(const char ∗)**. This is unnecessary anyway, since the prototypes in **<libintl.h>** already specify this type.