NAME

wninit, re_wninit, cntwords, strtolower, ToLowerCase, strsubst, getptrtype, getpos, getsstype, StrToPos, GetSynsetForSense, GetDataOffset, GetPolyCount, WNSnsToStr, GetValidIndexPointer, GetWNSense, GetSenseIndex, default_display_message

SYNOPSIS

#include "wn.h"

int wninit(void);

int re_wninit(void);

int cntwords(char *str, char separator);

char *strtolower(char *str);

char *ToLowerCase(char *str);

char *strsubst(char *str, char from, char to);

int getptrtype(char *ptr_symbol);

int getpos(char *ss_type);

int getsstype(char *ss_type);

int StrToPos(char pos);

SynsetPtr GetSynsetForSense(char *sense_key);

long GetDataOffset(char *sense_key);

int GetPolyCount(char *sense_key);

char *WNSnsToStr(IndexPtr idx, int sense_num);

IndexPtr GetValidIndexPointer(char *str, int pos);

int GetWNSense(char *lemma, *lex_sense);

SnsIndexPtr GetSenseIndex(char *sense_key);

int GetTagcnt(IndexPtr idx, int sense);

int default_display_message(char *msg);

DESCRIPTION

The WordNet library contains many utility functions used by the interface code, other library functions, and various applications and tools. Only those of importance to the WordNet search code, or which are generally useful are described here.

wninit( ) opens the files necessary for using WordNet with the WordNet library functions. The database files are opened, and **morphinit( )** is called to open the exception list files. Returns **0** if successful, **-1** otherwise. The database and exception list files must be open before the WordNet search and morphology functions are used. If the database is successfully opened, the global variable **OpenDB** is set to **1**. Note that it is possible for the database files to be opened (**OpenDB == 1**), but not the exception list files.

re_wninit( ) is used to close the database files and reopen them, and is used exclusively for WordNet development. **re_morphinit( )** is called to close and reopen the exception list files. Return codes are as described above.

cntwords( ) counts the number of underscore or space separated words in *str*. A hyphen is passed in *separator* if is is to be considered a word delimiter. Otherwise *separator* can be any other character, or an underscore if another character is not desired.

strtolower( ) converts *str* to lower case and removes a trailing adjective marker, if present. *str* is actually modified by this function, and a pointer to the modified string is returned.

ToLowerCase( ) converts *str* to lower case as above, without removing an adjective marker.

strsubst( ) replaces all occurrences of *from* with *to* in *str* and returns resulting string.

getptrtype( ) returns the integer *ptr_type* corresponding to the pointer character passed in *ptr_symbol*. See **wnsearch**(3WN) for a table of pointer symbols and types.

getpos( ) returns the integer constant corresponding to the synset type passed. *ss_type* may be one of the following: **n, v, a, r, s**. If **s** is passed, **ADJ** is returned. Exits with **-1** if *ss_type* is invalid.

getsstype( ) works like **getpos( )**, but returns **SATELLITE** if *ss_type* is **s**.

StrToPos( ) returns the integer constant corresponding to the syntactic category passed in *pos*. *string* must be one of the following: **noun, verb, adj, adv**. **-1** is returned if *pos* is invalid.

GetSynsetForSense( ) returns the synset that contains the word sense *sense_key* and **NULL** in case of error.

GetDataOffset( ) returns the synset offset for synset that contains the word sense *sense_key*, and **0** if *sense_key* is not in sense index file.

GetPolyCount( ) returns the polysemy count (number of senses in WordNet) for *lemma* encoded in *sense_key* and **0** if word is not found.

WNSnsToStr( ) returns sense key encoding for *sense_num* entry in *idx*.

GetValidIndexPointer( ) returns the Index structure for *word* in *pos*. Calls **morphstr**(3WN) to find a valid base form if *word* is inflected.

GetWNSense( ) returns the WordNet sense number for the sense key encoding represented by *lemma* and *lex_sense*.

GetSenseIndex( ) returns parsed sense index entry for *sense_key* and **NULL** if *sense_key* is not in sense index.

GetTagcnt( ) returns the number of times the sense passed has been tagged according to the *cntlist* file.

default_display_message( ) simply returns **-1**. This is the default value for the global variable **display_message**, that points to a function to call to display an error message. In general, applications (including the WordNet interfaces) define an application specific function and set **display_message** to point to it.

NOTES

include/wn.h lists all the pointer and search types and their corresponding constant values. There is no description of what each search type is or the results returned. Using the WordNet interface is the best way to see what types of searches are available, and the data returned for each.

SEE ALSO

wnintro(3WN), **wnsearch**(3WN), **morph**(3WN), **wnintro**(5WN), **wnintro**(7WN).

WARNINGS

Error checking on passed arguments is not rigorous. Passing **NULL** pointers or invalid values will often cause an application to die.