Complete SDK Functionality List

Functions available in AnNakel SDK

1 -Set Options:

bool SetTransliterationEa(int iOption);
iOption = EA_OPTION_NO_TRANSLIT No transliteration
iOption = EA_OPTION_TRANSLIT Transliterates Unknown Words
iOption = EA_OPTION_TRANSLIT_ABBREV Transliterates Unknown Words and abbreviations

bool SetCapConversionEa(int iOption);
iOption = EA_CAP_NO_CONVERSION The capitals are not converted in lower case. AnNakel is looking for an exact matching between the source text and the dictionary
iOption = EA_CAP_CONVERTED The capitals are converted in lower case characters

bool SetArabicVowelsEa(int iOption);
iOption = EA_ARABIC_NO_VOWELS The Arabic Target text is not vocalized
iOption = EA_ARABIC_WITH_VOWELS The Arabic Target text is vocalized

bool SetMasderMuawalEa(int iOption);
iOption = EA_MASDER_MUAWAL Verbal phrase for English Infinitive phrase.
iOption = EA_NO_MASDER_MUAWAL Noun phrase for English Infinitive phrase.

bool SetNoTranslationEa(char cStart, char cStop);
When this option is Set, the English Source Text located between cStart and cStop will not be
translated.
For example if cStart = '[' and cStop = ']', and you ask to translate the sentence:
"Dr Ahmed is a graduate of [Brown University] ",
the words " Brown University " will stay in English in the target translation in Arabic.

bool SetNoTranslationAe(char cStart, char cStop);
When this option is Set, the Arabic Source Text located between cStart and cStop will not be translated.

If the return code of an AnNakel SDK function is FALSE, call the Windows function GetLastError(); and refer to the error codes in Appendix 1.

2 -Get Options:

int GetTransliterationEa(void);
Retrieve the Transliteration State.
Return value are: (option.h)
EA_OPTION_NO_TRANSLIT No transliteration
EA_OPTION_TRANSLIT Transliterates Unknown Words
EA_OPTION_TRANSLIT_ABBREV Transliterates Unknown Words and abbreviations

int GetCapConversionEa(void);
Retrieve the CapConversion State.
Return value are: (option.h)
EA_CAP_NO_CONVERSION AnNakel is looking for an exact matching between the source text and the dictionary
EA_CAP_CONVERTED The capitals are converted in lower case characters

int GetArabicVowelsEa(void);
Retrieve the ArabicVowels State.
Return value are: (option.h)
EA_ARABIC_NO_VOWELS The Arabic Target text is not vocalized
EA_ARABIC_WITH_VOWELS The Arabic Target text is vocalized

int GetMasderMuawalEa(void);
Retrieve the Masder Muawal State.
Return value are: (option.h)
EA_MASDER_MUAWAL Verbal phrase for English Infinitive phrase.
EA_NO_MASDER_MUAWAL Noun phrase for English Infinitive phrase.

bool GetNoTranslationEa(char *cStart, char *cStop);
Retrieve the cStart and cStop characters witch delimite the beginning and the end of a non translation part in the English Source text.

bool GetNoTranslationAe(char *cStart, char *cStop);
Retrieve the cStart and cStop characters witch delimite the beginning and the end of a non translation part in the Arabic Source text.

3 -Translation functions from Source Text to Target Text

int NakelTransEa(char *lpSrc, char *lpTrg);

Translate from English to Arabic. The source text (English) is in lpSrc and the target text (Arabic) is
in lpTrg.
The source text is 8 bits coded or Unicode. If source text is 8 bits, Arabic Target text is coded in
cp1256, if source text is Unicode (2 char), target text is Unicode (2 char).
The return value is the lenght of the source text which has been translated.
If Zero, call the Windows function GetLastError();
The error codes are listed in the Appendix 1 "Error codes"

int NakelTransAe(char *lpSrc, char *lpTrg);

Translate from Arabic to English. The source text (Arabic) is in lpSrc and the target text (English) is
in lpTrg.
The source text is 8 bits coded or Unicode. If source text is in cp1256, Arabic Target text is Ansi coded
if source text is Unicode (2 bytes), target text is Unicode (2 bytes).
The return value is the lenght of the source text which has been translated.
If Zero, call the Windows function GetLastError();
The error codes are listed in the Appendix 1 "Error codes"

bool TranslateOneFileTxt(char *FileName, char cSens);

Translate the 'txt' file FileName. If cSens is egual to 'E', the source text is English and the target is
Arabic. If cSens is egual to 'A', the source text is Arabic and the target is English.
The source file is 8 bits coded or Unicode. If source file is 8 bit coded, target file is 8 bit coded.
If source file is Unicode (2 bytes), target file is Unicode (2 bytes), the two first characters are 0xFF
and 0xFE.
If the return value is FALSE, call the Windows function GetLastError();
The error codes are listed in the Appendix 1 "Error codes"

bool TranslateAllFileTxt(char *DirName, char cSens);

Translate all the files with 'txt' extension located in the directory DirName.
If cSens is egual to 'E', the source text is English and the target is Arabic.
If cSens is egual to 'A', the source text is Arabic and the target is English.
The source file is 8 bits coded or Unicode. If source file is 8 bit coded, target file is 8 bit coded.

If source file is Unicode (2 bytes), target file is Unicode (2 bytes), the two first characters are 0xFF
and 0xFE.
If the return value is FALSE, call the Windows function GetLastError();
The error codes are listed in the Appendix 1 "Error codes"

bool TranslateOneFileRtf(char *FileName, char cSens);

Translate the 'rtf' file FileName. If cSens is egual to 'E', the source text is English and the target is
Arabic. If cSens is egual to 'A', the source text is Arabic and the target is English.
If the return value is FALSE, call the Windows function GetLastError();
The error codes are listed in the Appendix 1 "Error codes"

bool TranslateAllFileRtf(char *DirName, char cSens);

Translate all the files with 'rtf' extension located in the directory DirName.
If cSens is egual to 'E', the source text is English and the target is Arabic.
If cSens is egual to 'A', the source text is Arabic and the target is English.
If the return value is FALSE, call the Windows function GetLastError();
The error codes are listed in the Appendix 1 "Error codes"

4 - Morphological Analyse function

int NakelMorpho(char *lpSrc, char **lpTrg, 'A', int iCode);

Provides a morphological analyse of each Arabic Word of the Source text in lpSrc and return the analyse in lpTrg. It splits each word in its components.
lpTrg is a pointer to a buffer defined by AnNakel.dll
iCode : the source text is 8 bits (cp1256) coded or Unicode.
#define CP1256 1
#define UNICODE 2
'A' means that the source text is Arabic.
The return value is the lenght of the analysed text.
If Zero, call the Windows function GetLastError();
The error codes are listed in the Appendix 1 "Error codes"

5 - Vocalisation function

int NakelVocalise(char *lpSrc, char **lpTrg, 'A', int iCode);

Vocalises the Arabic Source text (puts arabic vowels and diacritics in each word) in lpSrc and return the analyse in lpTrg.
lpTrg is a pointer to a buffer defined by AnNakel.dll
iCode : the source text is 8 bits (cp1256) coded or Unicode.
#define CP1256 1
#define UNICODE 2
'A' means that the source text is Arabic.
The return value is the lenght of the vocalised text.
If Zero, call the Windows function GetLastError();
The error codes are listed in the Appendix 1 "Error codes"

6 - Conjugation function

int NakelConjuguate(char *lpSrc, char **lpTrg, 'A', int iOptions);

Conjugates the Arabic verb in lpSrc and return the analyse in lpTrg.
lpTrg is a pointer to a buffer defined by AnNakel.dll
iOptions : The Mode and Tense of the conjugation.
'A' means that the conjugation is Arabic.
The return value is the lenght of the conjugation.
If Zero, call the Windows function GetLastError();
The error codes are listed in the Appendix 1 "Error codes"

7 - Stop function

bool NakelStopTranslate(int * iNbSentence, int *iPercent;
Stop the translation of files or directories.
iNbSentence is an integer where is returned the number of sentences translated
iPercent is an integer between 0 and 100 corresponding to the ratio between of the text translated and
the global amount of text.
If the return value is FALSE, call the Windows function GetLastError();
The error codes are listed in the Appendix 1 "Error codes"

8 - Get Number of Sentence

bool NakelGetNumberOfSentence(int * iNbSentence, int *iPercent;
iNbSentence is an integer where is returned the number of sentences translated during translation of a
file or directory.
iPercent is an integer between 0 and 100 corresponding to the ratio between of the text translated and
the global amount of text.
If the return value is FALSE, call the Windows function GetLastError();
The error codes are listed in the Appendix 1 "Error codes"