

OpenTM2 for Windows

API Call Reference

Version 1.2.2

Content

Overview and terminology	4
Data types	5
Sample code	6
EqfAddCTIDList	7
EqfAnalyzeDoc	8
EqfAnalyzeDocEx	10
EqfArchiveTM	12
EqfBuildSegDocName	13
EqfChangeFolProps	14
EqfChangeFolPropsEx	15
EqfChangeMFlag	16
EqfCheckSpelling	17
EqfCleanMemory	18
EqfClearMTFlag	19
EqfCountWords	20
EqfCountWordsInString	21
EqfCreateCntReport	22
EqfCreateControlledFolder	24
EqfCreateCountReport	27
EqfCreateFolder	29
EqfCreateITM	31
EqfCreateMarkup	33
EqfCreateSubFolder	35
EqfDeleteDict	37
EqfDeleteFolder	38
EqfDeleteMem	39
EqfDeleteMTLog	40
EqfDictionaryExists	41
EqfDocumentExists	42
EqfEndSession	43
EqfExportDict	44
EqfExportDoc	45
EqfExportFolder	46
EqfExportFolderFP	47
EqfExportFolderFPas	48
EqfExportMem	49
EqfExportSegs	50
EqfFilterNoMatchFile	51
EqfFolderExists	53
EqfFreeSegFile	54

EqfGetFolderProp	55
EqfGetLastError.....	56
EqfGetMatchLevel.....	57
EqfGetProgress	58
EqfGetSegmentNumber	59
EqfGetSegNum	60
EqfGetSegW	61
EqfGetShortName	62
EqfGetSourceLine	63
EqfGetSysLanguage	64
EqfGetVersionEx	65
EqfImportDict	66
EqfImportDoc	67
EqfImportFolder	69
EqfImportFolderFP	70
EqfImportMem	71
EqfLoadSegFile	72
EqfOpenDoc	73
EqfOpenDocEx	74
EqfOrganizeMem.....	75
EqfProcessNomatch	76
EqfProcessNomatchEx	77
EqfReduceToStemForm	79
EqfRemoveDocs	80
EqfRename.....	81
EqfRestoreDocs	82
EqfSetSysLanguage	83
EqfSimpleGetmatchLevel	84
EqfStartSession	85
EqfUpdateSegW	86
EqfWriteSegFile	87

Overview and terminology

Each OpenTM2 function includes a generic data block, which is encapsulated in the session handle. This session handle is created by the *EqfStartSession* call. It ensures that several OpenTM2 functions can run concurrently. The functions are delivered as a library and a dynamic-link library (DLL) following the standard PASCAL calling conventions. The include file OTMFUNC.H contains the prototypes of all available functions.

The long-running tasks, such as the export or the organization of a Translation Memory, are split into small units of work. The return code indicates if the task has completed successfully or if data is pending. The calling application must allocate the memory and free it when no longer used. In this way, the interface is independent of any compiler or runtime libraries used.

Data types

The API call interface for OpenTM2 functions uses the following data types for parameters and return codes:

HSESSION	The session handle that is created by <i>EqfStartSession</i> . It must be specified in all other functions of the non-DDE interface.
PHSESSION	The pointer to a HSESSION variable.
LONG	A long (32-bit) signed integer. In the non-DDE interface, this data type is used for option flags. Use 0L if no options are to be specified.
PSZ	The pointer to a zero-terminated string (C-language string). Use NULL if no parameter is specified.
USHORT	A short (16-bit) unsigned integer value. This data type is used for return codes.
PUSHORT	The pointer to a variable of type USHORT.

Sample code

The following sample is written in C. It shows how to create a new folder and how to import and analyze documents.

```
USHORT usRC = 0;
HSESSION hSession = 0L;

// start the Eqf session usRC =
EqfStartSession( &hSession );

// create the folder SAMPLE1
if ( !usRC )
{ usRC = EqfCreateFolder( hSession, "SAMPLE1", NULL, '\0', "MEM1", "EQFASCII",
    NULL, NULL, "English(U.S.)",
                                "German(national)" );
}

// import the documents TEST1.DOC and TEXT2.DOC into folder SAMPLE1 if ( !usRC )
{ do { usRC = EqfImportDoc( hSession, "SAMPLE1", NULL,
    "C:\\TEXT1.DOC,C:\\TEXT2.DOC",
    NULL, NULL, NULL, NULL, NULL, NULL, 0L );
    } while( usRC == CONTINUE_RC );
}

// Analyze all documents of folder SAMPLE1
if ( !usRC )
{ do { usRC = EqfAnalyzeDoc( hSession, "SAMPLE1", NULL, NULL, 0L );
    } while( usRC == CONTINUE_RC );
}

// end the Eqf session
if ( hSession != 0L )
{
    EqfEndSession( hSession );
}
```

EqfAddCTIDList

```
USHORT EqfAddCTIDList ( HSESSION  hSession,  
  
                        PSZ         pszFolderName,  
  
                        PSZ         pszCTIDListFile  
  
                        )
```

Associate a HAMSTER/Global memory CTID options file to a folder.

Parameters:

- hSession* the session handle returned by the EqfStartSession call
- pszFolderName* name of the folder
- pszCTIDListFile* fully qualified name of the CTID list file

Returns:

0 if successful or an error code in case of failures

EqfAnalyzeDoc

```
USHORT EqfAnalyzeDoc ( HSESSION hSession,
                        PSZ pszFolderName,
                        PSZ pszDocuments,
                        PSZ pszMemname,
                        LONG lOptions
                      )
```

Analyze one or more documents.

Parameters:

hSession	the session handle returned by the EqfStartSession call
pszFolderName	name of folder
pszDocuments	comma separated list with document names or NULL to analyze all documents of the folder
pszMemname	Translation Memory to be used for the analysis
lOptions	<div>analysis options<ul style="list-style-type: none">• TMMATCH_OPT prepare memory match word count• ADDTOMEM_OPT write found proposals to folder/document Translation Memory• AUTOSUBST_OPT auto-substitute memory matches• UNTRANSLATED_OPT write segments without proposals to an external memory (EXP)• AUTOLAST_OPT when more exact proposals are available use newest proposal for auto-substitution• AUTOJOIN_OPT auto-join segments• AUTOCONTEXT_OPT when more exact proposals are available use proposal from same document for auto-substitution• REDUNDCOUNT_OPT perform redundancy counting• IGNOREPATH_OPT ignore path of document names when looking for exact matches• ADJUSTLEADWS_OPT adjust leading white space of auto-substituted segments• ADJUSTTRAILWS_OPT adjust trailing white space of auto-substituted segments• RESPECTCRLF_OPT do not auto-substitute segments when number of line breaks are different• NOBLANKATSEGEND_OPT remove blank following a period at the end of the segment• NOSUBSTIFIDENTICAL_OPT do not auto-substitute segments when source and target of the proposal is identical• PROTECTXMPSCREEN_OPT protect xmp, screen, msgnum, meta, and codeblock sections• SENDTOMT_OPT send to / prepare data for MT systems• STOPATFIRSTEXACT_OPT stop searching Translation Memory databases at first exact match found• IGNORECOMMENTED_OPT ignore memory proposals with comments in the additional info area• PROTECTXMP_OPT protect xmp sections• PROTECTMSGNUM_OPT protect msgnum sections• PROTECTMETA_OPT protect meta sections• PROTECTSCREEN_OPT protect screen sections• PROTECTCODEBLOCK_OPT protect codeblock sections• OVERWRITE_OPT overwrite documents in</div>

Returns:

0 if successful or an error code in case of failures

EqfAnalyzeDocEx

USHORT EqfAnalyzeDocEx	(HSESSION	hSession,
		PSZ	pszFolderName,
		PSZ	pszDocuments,
		PSZ	pszMemname,
		PSZ	pszProfile,
		PVOID	pvReserved,
		LONG	lOptions
)		

Analyze one or more documents (with additional parameters)

Parameters:

hSession	the session handle returned by the EqfStartSession call
pszFolderName	name of folder
pszDocuments	comma separated list with document names or NULL to analyze all documents of the folder
pszMemname	Translation Memory to be used for the analysis
pszProfile	analysis profile to be used for the analysis or NULL if not used
pvReserved	reserved for future enhancements, has to be NULL
lOptions	<div>analysis options<ul style="list-style-type: none">• TMMATCH_OPT prepare memory match word count• ADDTOMEM_OPT write found proposals to folder/document Translation Memory• AUTOSUBST_OPT auto-substitute memory matches• UNTRANSLATED_OPT write segments without proposals to an external memory (EXP)• AUTOLAST_OPT when more exact proposals are available use newest proposal for auto-substitution• AUTOJOIN_OPT auto-join segments• AUTOCONTEXT_OPT when more exact proposals are available use proposal from same document for auto-substitution• REDUNDCOUNT_OPT perform redundancy counting• IGNOREPATH_OPT ignore path of document names when looking for exact matches• ADJUSTLEADWS_OPT adjust leading white space of auto-substituted segments• ADJUSTTRAILWS_OPT adjust trailing white space of auto-substituted segments• RESPECTCRLF_OPT do not auto-substitute segments when number of line breaks are different• NOBLANKATSEGENG_OPT remove blank following a period at the end of the segment• NOSUBSTIFIDENTICAL_OPT do not auto-substitute segments when source and target of the proposal is identical• PROTECTXMPSCREEN_OPT protect xmp, screen, msgnum, meta, and codeblock sections• SENDTOMT_OPT send to / prepare data for MT systems• STOPATFIRSTEXACT_OPT stop searching Translation Memory databases at first exact match found</div>

- IGNORECOMMENTED_OPT ignore memory proposals with comments in the additional info area
- PROTECTXMP_OPT protect xmp sections
- PROTECTMSGNUM_OPT protect msgnum sections
- PROTECTMETA_OPT protect meta sections
- PROTECTSCREEN_OPT protect screen sections
- PROTECTCODEBLOCK_OPT protect codeblock sections
- OVERWRITE_OPT overwrite documents in

Returns:

0 if successful or an error code in case of failures

EqfArchiveTM

```
USHORT EqfArchiveTM ( HSESSION hSession,
                      PSZ pszFolderName,
                      CHAR chToDrive,
                      PSZ pszDocuments,
                      PSZ pszMemname,
                      LONG lOptions
                    )
```

Create a archive memory from translated documents.

Parameters:

hSession	the session handle returned by the EqfStartSession call
pszFolderName	name of the folder
chToDrive	folder drive, can be ignored
pszDocuments	comma separated list with document names or NULL to include all folder documents
pszMemname	name of output Translation Memory
lOptions	options for archive memory function <ul style="list-style-type: none">OVERWRITE_OPT delete all existing proposals in the Translation Memory before startingUSEASFOLDERTM_OPT use specified Translation Memory as folder memorySOURCESOURCEMEM_OPT create a source-source Translation MemorySETMFLAG_OPT set machine translation flag in the proposals written to the Translation Memory

Returns:

0 if successful or an error code in case of failures

EqfBuildSegDocName

```
USHORT EqfBuildSegDocName ( HSESSION  hSession,
                             PSZ        pszFolderName,
                             PSZ        pszDocumentName,
                             USHORT     fSource,
                             PSZ        pszSegFile
                             )
```

Construct the fully qualified name of a segmented document from long folder name and document name.

Parameters:

hSession	the session handle returned by the EqfStartSession call
pszFolderName	pointer to name of the folder
pszDocumentName	pointer to document name
fSource	when TRUE the name of the segmented source file name is constructed, when FALSE the name of the segmented target file name is constructed
pszSegFile	points to buffer receiving the fully qualified document name

Returns:

0 if successful or an error code in case of failures

EqfChangeFolProps

```
USHORT EqfChangeFolProps ( HSESSION  hSession,

                           PSZ         pszFolderName,

                           CHAR        chTargetDrive,

                           PSZ         pszTargetLanguage,

                           PSZ         pszMemName,

                           PSZ         pszDictionaries

                           )
```

Change the properties of a folder.

Parameters:

<i>hSession</i>	the session handle returned by the EqfStartSession call
<i>pszFolderName</i>	pointer to the buffer containing the folder name
<i>chTargetDrive</i>	the drive letter of the drive the folder resides on (is not required anymore and should be left empty)
<i>pszTargetLanguage</i>	pointer to the buffer containing the new folder target language (OpenTM2 language name) or NULL when the target language should not be changed
<i>pszMemName</i>	pointer to the buffer containing the name of the new folder Translation Memory or NULL when the target language should not be changed
<i>pszDictionaries</i>	pointer to the buffer containing the comma separated new list of dictionaries or NULL if dictionaries should not be changed

Returns:

0 if successful or an error code in case of failures

EqfChangeFolPropsEx

```
USHORT EqfChangeFolPropsEx ( HSESSION  hSession,

                             PSZ         pszFolderName,

                             CHAR        chTargetDrive,

                             PSZ         pszTargetLanguage,

                             PSZ         pszMemName,

                             PSZ         pszDictionaries,

                             PSZ         pszROMemories,

                             PSZ         pszDescription,

                             PSZ         pszProfile,

                             PSZ         pszUnUsed2

                             )
```

Change the properties of a folder (extended version)

Parameters:

hSession	the session handle returned by the EqfStartSession call
pszFolderName	pointer to the buffer containing the folder name
chTargetDrive	the drive letter of the drive the folder resides on (is not required anymore and should be left empty)
pszTargetLanguage	pointer to the buffer containing the new folder target language (OpenTM2 language name) or NULL for no change
pszMemName	pointer to the buffer containing the name of the new folder Translation Memory or NULL for no change
pszDictionaries	pointer to the buffer containing the comma separated new list of dictionaries or NULL for no change
pszROMemories	pointer to the buffer containing the new list of read-only Translation Memory databases or NULL for no change
pszDescription	pointer to the buffer containing the new folder description or NULL for no change
pszProfile	pointer to the buffer containing the new calculation profile name or NULL for no change
pszUnUsed2	parameter currently not in use and reserved for future enhancements, should be NULL

Returns:

0 if successful or an error code in case of failures

EqfChangeMFlag

```
USHORT EqfChangeMFlag ( HSESSION  hSession,
                        PSZ         pszMemName,
                        LONG         lAction
                        )
```

Create a new folder.

Parameters:

- hSession

the session handle returned by the EqfStartSession call
- pszMemName

name of the Translation Memory
- type of action
- lAction

- CLEAR_MMOPT clear the machine translation flag
 - SET_MMOPT set the machine translation flag

Returns:

0 if successful or an error code in case of failures

EqfCheckSpelling

```
unsigned short EqfCheckSpelling      ( HSESSION   hSession,  
  
                                     char *      pszLanguage,  
  
                                     char *      pszInputTerms,  
  
                                     char *      pszInputFile,  
  
                                     char *      pszReport,  
  
                                     LONG        lOptions  
  
                                     )
```

Check the spelling of a list of words.

Parameters:

<i>hSession</i>	OpenTM2 session handle returned by EqfStartSession
<i>pszLanguage</i>	name of the language being used for the spell checking
<i>pszInputTerms</i>	a comma separated list of terms or NULL if a input file is being used
<i>pszInputFile</i>	the fully qualified name of a plain text file containing the terms, one term per line or NULL if <i>pszInputTerms</i> is being used
<i>pszReport</i>	name of the report file receiving the results of the operation
<i>lOption</i>	options for the output of the report: TEXT_OUTPUT_OPT for plain text output (CSV) or XML_OUTPUT_OPT (default) for XML output

Returns:

0 if successful or an error code

EqfCleanMemory

```
USHORT EqfCleanMemory ( HSESSION hSession,
                        PSZ pszFolder,
                        PSZ pszInMemory,
                        PSZ pszOutMemory,
                        LONG lOptions
                      )
```

Remove irrelevant (for the given folder) proposals from an external memory The relevant segments are either stored in an internal or external memory. The external memory format is EXP in UTF-16 encoding.

Parameters:

hSession	the session handle returned by the EqfStartSession call
pszFolder	name of folder containing the translatable material
pszInMemory	fully qualified name of external input memory (EXP format, encoding: UTF-16)
pszOutMemory	name of internal output memory or fully qualified name of external output memory
lOptions	options for processing <ul style="list-style-type: none">CLEANMEM_INTERNAL_MEMORY_OPT write the relevant proposals to an internal memoryCLEANMEM_EXTERNAL_MEMORY_OPT write the relevant proposals to an external memoryCOMPLETE_IN_ONE_CALL_OPT complete the process in one call, without this option the function processes a small unit and returns to the calling process with a return code of CONTINUE_RC, the calling process has to re-call the API call until any other return code is returnedCLEANMEM_LOGGING_OPT activate logging (will append log info to the log file .LOG)CLEANMEM_BESTMATCH_OPT when specified only the best match will be written to the output memory, without this option the best 3 matches will be written to the output memoryCLEANMEM_MERGE_OPT merge the relevant proposal into an existing internal memory, without this option the memory is cleared before adding the relevant proposalsCLEANMEM_KEEP_DUPS_OPT keep duplicate exact matches and fuzzy matches in the memory

Returns:

0 if successful, CONTINUE_RC to indicate that the processing is not finished yet, or an error code in case of failures

EfqClearMTFlag

```
unsigned short EqfClearMTFlag ( HSESSION hSession,  
  
                                char * pszInMemory,  
  
                                char * pszOutMemory  
  
                                )
```

Clears the MT flag of an external memory in the EXP format.

This API function processes a memory in the .EXP format (encoding UTF-16, ANSI. or ASCII) and clears any machine translation flag (MT flags) of the memory proposals. When an output memory is specified the processed memory is written to the specified output file otherwise the input memory is overwritten with the modified memory.

Parameters:

<i>pszInMemory</i>	fully qualified file name of the input memory
<i>pszOutMemory</i>	fully qualified file name of the output memory, if not specified the output memory overwrites the input memory

Returns:

0 if successful or an error code in case of failures

EqfCountWords

```
USHORT EqfCountWords ( HSESSION  hSession,

                      PSZ         pszFolderName,

                      PSZ         pszDocuments,

                      LONG        lOptions,

                      PSZ         pszOutFile

                      )
```

Count the number of words in documents.

This API call counts the words of the documents in a folder or of a subset of the documents in a folder. The word count report can be a source word count (SOURCE_OPT), a target word count (TARGET_OPT), a duplicate word count (DUPLICATE_OPT), or a memory match word count (TMMATCH_OPT) The format of the word count report can be either text format (TEXT_OUTPUT_OPT), HTML format (HTML_OUTPUT_OPT), or XML format (XML_OUTPUT_OPT).

Parameters:

hSession	OpenTM2 session handle returned by EqfStartSession
pszFolderName	the name of the folder containing the documents being counted
pszDocuments	list of documents or NULL if all documents of the folder are to be counted
lOptions	<div>options for the word counting<ul style="list-style-type: none">• TMMATCH_OPT create a memory match word count• DUPLICATE_OPT create a duplicates word count• SOURCE_OPT create a source word count• TARGET_OPT create a target word count• FUZZYMATCH_OPT include fuzzy match information in memory match word count• SEPERATEREPLMATCH_OPT count replace matches separately in memory match word count• DUPMEMMATCH_OPT include memory match information in duplicates word count• TEXT_OUTPUT_OPT create a plain text word count• XML_OUTPUT_OPT create a word count in XML format• HTML_OUTPUT_OPT create a word count in HTML format</div>
pszOutFile	fully qualified name of the file receiving the word count output

Returns:

0 if successful or an error code

EqfCountWordsInString

```
unsigned short EqfCountWordsInString ( HSESSION      hSession,  
  
                                       char *         pszMarkup,  
  
                                       char *         pszLanguage,  
  
                                       wchar_t *      pszText,  
  
                                       unsigned long * pulWords,  
  
                                       unsigned long * pulInlineTags  
  
                                       )
```

Count the number of words in a given string.

Parameters:

<i>hSession</i>	OpenTM2 session handle returned by EqfStartSession
<i>pszMarkup</i>	name of the markup table to be used for the recognition of in-line tags, if this parameter is NULL no in-line tag recognition will be performed
<i>pszLanguage</i>	OpenTM2 name for the language of the given text
<i>pszText</i>	null-terminated string containing the text to be counted, the encoding is UTF-16
<i>pulWords</i>	points to an unsigned long value receiving the number of words in the text
<i>pulInlineTags</i>	points to an unsigned long value receiving the number of inline tags in the text

Returns:

0 if successful or an error code

EqfCreateCntReport

```
USHORT EqfCreateCntReport ( HSESSION      hSession,
                             CHAR          chDriveLetter,
                             PSZ           pszFolderName,
                             PSZ           pszDocuments,
                             PREPORTTYPE   pReportType,
                             PSZ           pszOutfileName,
                             PSZ           pszFormat,
                             PSZ           pszProfile,
                             PREPORTSETTINGS pRepSettings,
                             PFACTSHEET    pFactSheet,
                             USHORT        usColumn,
                             USHORT        usCategory,
                             PFINALFACTORS pFinalFactors,
                             LONG          lOptSecurity,
                             BOOL          bSingleShipment
                             )
```

Create a counting report (old interface, please use EqfCreateCountReport instead)

Parameters:

hSession	the session handle returned by the EqfStartSession call
chDriveLetter	drive letter of the folder location, not used anymore and should be empty
pszFolderName	name of the folder
pszDocuments	comma separated list of document names or NULL if complete folder should be counted
pReportType	pointer to a REPORTTYPE structure containing the report type information
pszOutfileName	fully qualified name of the output report file
pszFormat	format of the output file <ul style="list-style-type: none">"ASCII" for a plain text report

- "HTML" for a HTML report
- "RTF" for a RTF report
- "XML" for a XML report

pszProfile	name of the calculation profile to be used, when specified the following parameters are not used and all information is retrieved from the profile instead
pRepSettings	pointer to a REPORTSETTINGS structure containing the report settings
pFactSheet	pointer to a FACTSHEET structure containing the factsheet infos
usColumn	number of columns to in the report
usCategory	number of categories in the report
pFinalFactors	pointer to a FINALFACTORS structure containing the payment factors
lOptSecurity	security check level
bSingleShipment	1 = count single shipments, 0 = count all shipments

Returns:

0 if successful or an error code in case of failures

EfqCreateControlledFolder

USHORT	EfqCreateControlledFolder	(HSESSION	<i>hSession,</i>
			PSZ	<i>pszFolderName,</i>
			PSZ	<i>pszDescription,</i>
			CHAR	<i>chTargetDrive,</i>
			PSZ	<i>pszMemname,</i>
			PSZ	<i>pszMarkup,</i>
			PSZ	<i>pszEditor,</i>
			PSZ	<i>pszDictionaries,</i>
			PSZ	<i>pszSourceLanguage,</i>
			PSZ	<i>pszTargetLanguage,</i>
			PSZ	<i>pszConversion,</i>
			PSZ	<i>pszReadOnlyMems,</i>
			PSZ	<i>pszPassword,</i>
			PSZ	<i>pszProjCoordName,</i>
			PSZ	<i>pszProjCoordMail,</i>
			PSZ	<i>pszTranslatorName,</i>
			PSZ	<i>pszTranslatorMail,</i>
			PSZ	<i>pszProductName,</i>
			PSZ	<i>pszProductFamily,</i>
			PSZ	<i>pszSimilarProduct,</i>
			PSZ	<i>pszProductDict,</i>
			PSZ	<i>pszProductMem,</i>
			PSZ	<i>pszPreviousVersion,</i>
			PSZ	<i>pszVersion,</i>

PSZ *pszShipmentNumber*

)

Create a new controlled/protected folder A controlled folder has some protected properties.

Parameters:

<i>hSession</i>	the session handle returned by the EqfStartSession call
<i>pszFolderName</i>	pointer to the buffer containing the folder name
<i>pszDescription</i>	pointer to the buffer containing the folder description or NULL if not used
<i>chTargetDrive</i>	the drive letter of the drive the folder should reside on, if left empty the folder is created on the OpenTM2 installation drive
<i>pszMemname</i>	pointer to the buffer containing the folder Translation Memory
<i>pszMarkup</i>	pointer to the buffer containing the name of the folder markup (e.g. IBMDITA)
<i>pszEditor</i>	pointer to the buffer containing the name of the editor to be used for documents of this folder
<i>pszDictionaries</i>	pointer to the buffer containing a comma separated list of dictionaries or NULL if no dictionaries should be used
<i>pszSourceLanguage</i>	pointer to the buffer containing the folder source language (OpenTM2 language name)
<i>pszTargetLanguage</i>	pointer to the buffer containing the folder target language (OpenTM2 language name)
<i>pszConversion</i>	pointer to the buffer containing the export conversion - currently not in use and should be NULL
<i>pszReadOnlyMems</i>	pointer to the buffer containing the a comma separated list of read only Translation Memory databases or NULL if not used
<i>pszPassword</i>	pointer to the buffer containing the password to be used for the folder
<i>pszProjCoordName</i>	pointer to the buffer containing the name of the project coordinator or NULL if not used
<i>pszProjCoordMail</i>	pointer to the buffer containing the project coordinator's mail or NULL if not used
<i>pszTranslatorName</i>	pointer to the buffer containing the name of the translator
<i>pszTranslatorMail</i>	pointer to the buffer containing the translator's mail
<i>pszProductName</i>	pointer to the buffer containing the name of the product
<i>pszProductFamily</i>	pointer to the buffer containing the product family
<i>pszSimilarProduct</i>	pointer to the buffer containing the similar product family
<i>pszProductDict</i>	pointer to the buffer containing the product subject area dictionary
<i>pszProductMem</i>	pointer to the buffer containing the product subject area memory

<code>pszPreviousVersion</code>	pointer to the buffer containing the previous version of the product
<code>pszVersion</code>	pointer to the buffer containing the Version of the product
<code>pszShipmentNumber</code>	pointer to the buffer containing the shipment number

Returns:

0 if successful or an error code in case of failures

EqfCreateCountReport

```
USHORT EqfCreateCountReport ( HSESSION  hSession,

                               PSZ        pszFolderName,

                               PSZ        pszDocuments,

                               PSZ        pszOutFile,

                               USHORT      usReport,

                               USHORT      usType,

                               PSZ        pszProfile,

                               LONG        lOptions

                               )
```

Create a counting report.

Parameters:

hSession	the session handle returned by the EqfStartSession call
pszFolderName	name of the folder containing the documents
pszDocuments	comma separated list of documents being counted or NULL if all documents of the folder should be counted
pszOutFile	fully qualified name of the output file
	ID of the report being created
usReport	<ul style="list-style-type: none">HISTORY_REP create a history reportCOUNTING_REP create a counting reportCALCULATING_REP create a calculating reportPREANALYSIS_REP create a pre-analysis reportREDUNDANCY_REP create a redundancy reportREDUNDANCYSEGMENT_REP create a redundant segment list
	type of the report being created valid for a history report are
usType	<ul style="list-style-type: none">BRIEF_SORTBYDATE_REPTYPE create a brief report sorted by dateBRIEF_SORTBYDOC_REPTYPE create a brief report sorted by documentsDETAIL_REPTYPE create a detailed report valid types for counting reportWITHTOTALS_REPTYPE counting report with totalsWITHOUTTOTALS_REPTYPE counting report without totals valid types for calculating report, pre-analysis report, redundancy report, and redundant segment listBASE_REPTYPE only baseBASE_SUMMARY_REPTYPE only summaryBASE_SUMMARY_FACTSHEET_REPTYPE base, summary, and factsheetSUMMARY_FACTSHEET_REPTYPE only summary and factsheetFACTSHEET_REPTYPE only factsheet

pszProfile	name of the counting report profile to be applied
	options for the counting report
	<ul style="list-style-type: none"> • OVERWRITE_OPT overwrite any existing output file
lOptions	one of the output format options
	<ul style="list-style-type: none"> • TEXT_OUTPUT_OPT create a plain text report • XML_OUTPUT_OPT create a XML report • HTML_OUTPUT_OPT create a HTML report

Returns:

0 if successful or an error code in case of failures

EqfCreateFolder

```
USHORT EqfCreateFolder ( HSESSION  hSession,

                        PSZ          pszFolderName,

                        PSZ          pszDescription,

                        CHAR          chTargetDrive,

                        PSZ          pszMemname,

                        PSZ          pszMarkup,

                        PSZ          pszEditor,

                        PSZ          pszDictionaries,

                        PSZ          pszSourceLanguage,

                        PSZ          pszTargetLanguage,

                        PSZ          pszConversion,

                        PSZ          pszReadOnlyMems

                        )
```

Create a new folder.

Parameters:

hSession	the session handle returned by the EqfStartSession call
pszFolderName	pointer to the buffer containing the folder name
pszDescription	pointer to the buffer containing the folder description or NULL if not used
chTargetDrive	the drive letter of the drive the folder should reside on, if left empty the folder is created on the OpenTM2 installation drive
pszMemname	pointer to the buffer containing the folder Translation Memory
pszMarkup	pointer to the buffer containing the name of the folder markup (e.g. IBMDITA)
pszEditor	pointer to the buffer containing the name of the editor to be used for documents of this folder
pszDictionaries	pointer to the buffer containing a comma separated list of dictionaries or NULL if no dictionaries should be used
pszSourceLanguage	pointer to the buffer containing the folder source language (OpenTM2 language name)

<code>pszTargetLanguage</code>	pointer to the buffer containing the folder target language (OpenTM2 language name)
<code>pszConversion</code>	pointer to the buffer containing the export conversion - currently not in use and should be NULL
<code>pszReadOnlyMems</code>	pointer to the buffer containing the a comma separated list of readonly Translation Memory databases or NULL if not used

Returns:

0 if successful or an error code in case of failures

EqfCreateITM

```
USHORT EqfCreateITM ( HSESSION  hSession,  
  
                     PSZ        pszMemName,  
  
                     PSZ        pszFilePairs,  
  
                     PSZ        pszMarkup,  
  
                     PSZ        pszSGMLMemFile,  
  
                     PSZ        pszSourceLanguage,  
  
                     PSZ        pszTargetLanguage,  
  
                     PSZ        pszSourceStartPath,  
  
                     PSZ        pszTargetStartPath,  
  
                     LONG       lType  
  
                     )
```

Create a Initial Translation Memory.

Parameters:

<i>hSession</i>	the session handle returned by the EqfStartSession call
<i>pszMemName</i>	name of the Translation Memory
<i>pszFilePairs</i>	list of file pairs as a comma separated list enclosed in parentheses
<i>pszMarkup</i>	markup table to be used
<i>pszSGMLMemFile</i>	external memory name or NUULL
<i>pszSourceLanguage</i>	Translation Memory source language
<i>pszTargetLanguage</i>	Translation Memory target language
<i>pszSourceStartPath</i>	startpath not stored for source files or NULL
<i>pszTargetStartPath</i>	startpath not stored for target files or NULL
	type of creation
<i>lType</i>	<ul style="list-style-type: none">• NOANA_TYP no analysis is necessary• NOTM_TYP no translation memory database• PREPARE_TYP only prepare the alignment

Returns:

0 if successful or an error code in case of failures

EqfCreateMarkup

```
USHORT EqfCreateMarkup ( HSESSION  hSession,  
  
                        PSZ         pszInFile,  
  
                        PSZ         pszOutFile  
  
                        )
```

Create a internal markup table (.TBL) from an external markup table (.TBX)

Parameters:

<i>hSession</i>	the session handle returned by the EqfStartSession call
<i>pszInFile</i>	fully qualified name of input file (.TBX format)
<i>pszOutFile</i>	fully qualified name of output file (.TBL format)

Returns:

0 if successful or an error code in case of failures

```
USHORT EqfCreateMem ( HSESSION  hSession,  
  
                     PSZ         pszMemname,  
  
                     PSZ         pszDescription,  
  
                     CHAR        chToDrive,  
  
                     PSZ         pszSourceLanguage,  
  
                     LONG        lOptions  
  
                     )
```

Create a new Translation Memory.

Parameters:

<i>hSession</i>	the session handle returned by the EqfStartSession call
<i>pszMemname</i>	name of the Translation Memory
<i>pszDescription</i>	description for new Translation Memory or NULL
<i>chToDrive</i>	target drive for new Translation Memory
<i>pszSourceLanguage</i>	Translation Memory source language
<i>lOptions</i>	type of new Translation Memory

- LOCAL_TM_OPT create a local memory using the EqfMemoryPlugin

Returns:

0 if successful or an error code in case of failures

EqfCreateSubFolder

```
USHORT EqfCreateSubFolder ( HSESSION  hSession,

                             PSZ        pszParentFolder,

                             PSZ        pszSubFolderName,

                             PSZ        pszMemName,

                             PSZ        pszMarkup,

                             PSZ        pszSourceLanguage,

                             PSZ        pszTargetLanguage,

                             PSZ        pszEditor,

                             PSZ        pszConversion,

                             PSZ        pszTranslator,

                             PSZ        pszTranslatorMail

                             )
```

Create a sub-folder.

Parameters:

hSession	the session handle returned by the EqfStartSession call
pszParentFolder	pointer to the buffer containing the name of the parent folder
pszSubFolderName	pointer to the buffer containing the name for the new sub-folder
pszMemname	pointer to the buffer containing the folder Translation Memory
pszMarkup	pointer to the buffer containing the name of the folder markup (e.g. IBMDITA)
pszSourceLanguage	pointer to the buffer containing the sub-folder source language (OpenTM2 language name)
pszTargetLanguage	pointer to the buffer containing the sub-folder target language (OpenTM2 language name)
pszEditor	pointer to the buffer containing the name of the editor to be used for documents of this sub-folder
pszConversion	pointer to the buffer containing the export conversion - currently not in use and should be NULL
pszTranslatorName	pointer to the buffer containing the name of the translator
pszTranslatorMail	pointer to the buffer containing the translator's mail

Returns:

0 if successful or an error code in case of failures

EqfDeleteDict

```
USHORT EqfDeleteDict ( HSESSION hSession,  
  
                       PSZ pszDictName  
  
                       )
```

Deletes the given dictionary.

Parameters:

hSession the session handle returned by the EqfStartSession call

pszDict name of the dictionary being deleted

Returns:

0 if successful or an error code in case of failures

```
USHORT EqfDeleteDoc ( HSESSION hSession,  
  
                      PSZ pszFolderName,  
  
                      PSZ pszDocuments  
  
                      )
```

Delete one or more documents.

Parameters:

hSession the session handle returned by the EqfStartSession call

pszFolderName name of the folder containing the documents

pszDocuments a comma separated list of documents being deleted

Returns:

0 if successful or an error code in case of failures

EqfDeleteFolder

```
USHORT EqfDeleteFolder ( HSESSION  hSession,
                          PSZ        pszFolderName
                          )
```

Delete a folder.

Parameters:

- hSession the session handle returned by the EqfStartSession call
- pszFolderName name of the folder being deleted

Returns:

0 if successful or an error code in case of failures

EqfDeleteMem

```
USHORT EqfDeleteMem ( HSESSION  hSession,  
  
                     PSZ        pszMemName  
  
                     )
```

Delete a Translation Memory.

Parameters:

- hSession* the session handle returned by the EqfStartSession call
- pszMemName* name of the memory being deleted

Returns:

0 if successful or an error code in case of failures

EqfDeleteMTLog

```
USHORT EqfDeleteMTLog ( HSESSION  hSession,  
  
                        PSZ        pszFolderName  
  
                        )
```

Delete the MT log files of a folder.

Parameters:

- hSession* the session handle returned by the EqfStartSession call
- pszFolderName* name of the folder

Returns:

0 if successful or an error code in case of failures

EqfDictionaryExists

```
USHORT EqfDictionaryExists ( HSESSION hSession,
                             PSZ pszDictionaryName
                             )
```

Check the existence of a dictionary.

Parameters:

- hSession* the session handle returned by the EqfStartSession call
- pszDictionaryName* name of the dictionary

Returns:

0 when the dictionary exists or an error code

EqfDocumentExists

```
USHORT EqfDocumentExists ( HSESSION  hSession,
                           PSZ         pszFolderName,
                           PSZ         pszDocumentName
                           )
```

Check the existence of a document.

Parameters:

- hSession the session handle returned by the EqfStartSession call
- pszDocumentName name of the document

Returns:

0 when the document exists or an error code

EqfEndSession

USHORT EqfEndSession (HSESSION *hSession*)

End a OpenTM2 API call session.

Parameters:

hSession the session handle returned by the EqfStartSession call

Returns:

0 if successful or an error code in case of failures

EqfExportDict

```
USHORT EqfExportDict ( HSESSION hSession,
                        PSZ pszDictName,
                        LONG lOptions,
                        PSZ pszOutFile
                      )
```

Export a dictionary.

Parameters:

hSession	the session handle returned by the EqfStartSession call
pszDictName	name of dictionary
lOptions	dictionary export options or 0L <ul style="list-style-type: none">OVERWRITE_OPT to overwrite existing files
pszOutFile	and one of the format options <ul style="list-style-type: none">ASCII_OPT import from SGML (ASCII encoded)ANSI_OPT import from SGML (ANSI encoded)UTF16_OPT import from SGML (Unicode UTF-16 encoded)DXT_UTF8_OPT import from DXT XML (UTF-8 encoded) fully qualified name of output file

Returns:

0 if successful or an error code in case of failures

EfqExportDoc

```
USHORT EqfExportDoc ( HSESSION hSession,  
  
                     PSZ pszFolderName,  
  
                     PSZ pszFiles,  
  
                     PSZ pszStartPath,  
  
                     LONG lOptions  
  
                     )
```

Export a document.

Parameters:

<i>hSession</i>	the session handle returned by the EqfStartSession call
<i>pszFolderName</i>	name of folder receiving the documents
<i>pszFiles</i>	list of input files (documents) in form of a comma separated list enclosed in parentheses
<i>pszStartPath</i>	optional start path
<i>lOptions</i>	<div>document import options or 0<ul style="list-style-type: none">• OVERWRITE_OPT to overwrite any existing fileand one of the export format options<ul style="list-style-type: none">• SOURCE_OPT export the document source file• TARGET_OPT export the document target (=translated) file• SNOMATCH_OPT export the list of segments with no matching memory proposal• VALFORMAT_ODT_OPT export as validation document (ODT format)• VALFORMAT_XML_OPT export as validation document (XML format)• VALFORMAT_HTML_OPT export as validation document (HTML format)• VALFORMAT_DOC_OPT export as validation document (MS DOC format)• PLAINXML_OPT export document as plain XML filefor the validation formats the following two additional options can specified<ul style="list-style-type: none">• VALFORMAT_COMBINE_OPT combine all documents into a single one• VALFORMAT_PROTSEGS_OPT include protected segments in the exported document</div>

Returns:

0 if successful or an error code in case of failures

EqfExportFolder

```
USHORT EqfExportFolder ( HSESSION hSession,
                          PSZ      pszFolderName,
                          CHAR      chTargetDrive,
                          LONG      lOptions,
                          PSZ      pszDocuments,
                          PSZ      pszDescription
                        )
```

Export a folder to the directory of the specified drive.

Parameters:

hSession	the session handle returned by the EqfStartSession call
pszFolderName	pointer to the buffer containing the name of folder being exported
chTargetDrive	folder export target drive
lOptions	options for the folder export or 0L <ul style="list-style-type: none">• DELETE_OPT delete the folder after successful export• WITHDICT_OPT add the dictionaries used by the folder• WITHMEM_OPT include the folder Translation Memory• WITHREADONLYMEM_OPT include the read-only Translation Memory databases• WITHDOCMEM_OPT include the Translation Memory databases specified in the document properties• OVERWRITE_OPT overwrite any existing exported folder• MASTERFOLDER_OPT export as master folder• XLIFF_OPT export as XLIFF folder
pszDocuments	pointer to the buffer containing the list of documents or NULL to export all documents of the folder
pszDescription	pointer to the buffer containing the export description or NULL if not used

Returns:

0 if successful or an error code in case of failures

EfqExportFolderFP

```
USHORT EfqExportFolderFP ( HSESSION hSession,  
  
                           PSZ      pszFolderName,  
  
                           PSZ      pszTargetPath,  
  
                           LONG     lOptions,  
  
                           PSZ      pszDocuments,  
  
                           PSZ      pszDescription  
  
                           )
```

Export a folder to the specified directory.

Parameters:

<i>hSession</i>	the session handle returned by the EfqStartSession call
<i>pszFolderName</i>	pointer to the buffer containing the name of folder being exported
<i>pszTargetPath</i>	pointer to the buffer containing target directory for the folder export
<i>lOptions</i>	<div>options for the folder export or 0L<ul style="list-style-type: none">• DELETE_OPT delete the folder after successful export• WITHDICT_OPT add the dictionaries used by the folder• WITHMEM_OPT include the folder Translation Memory• WITHREADONLYMEM_OPT include the read-only Translation Memory databases• WITHDOCMEM_OPT include the Translation Memory databases specified in the document properties• OVERWRITE_OPT overwrite any existing exported folder• MASTERFOLDER_OPT export as master folder• XLIFF_OPT export as XLIFF folder</div>
<i>pszDocuments</i>	pointer to the buffer containing the list of documents or NULL to export all documents of the folder
<i>pszDescription</i>	pointer to the buffer containing the export description or NULL if not used

Returns:

0 if successful or an error code in case of failures

EqfExportFolderFPas

```
USHORT EqfExportFolderFPas ( HSESSION hSession,
                             PSZ pszFolderName,
                             PSZ pszTargetPath,
                             PSZ pszExportAs,
                             LONG lOptions,
                             PSZ pszDocuments,
                             PSZ pszDescription,
                             PSZ pszMemoryExportAs
                             )
```

Export a folder to the specified directory using a different folder name.

Parameters:

hSession	the session handle returned by the EqfStartSession call
pszFolderName	pointer to the buffer containing the name of folder being exported
pszTargetPath	pointer to the buffer containing target directory for the folder export
pszExportAs	pointer to the buffer containing name to be use for the exported folder or NULL if not used
lOptions	<div>options for the folder export or 0L<ul style="list-style-type: none">• DELETE_OPT delete the folder after successful export• WITHDICT_OPT add the dictionaries used by the folder• WITHMEM_OPT include the folder Translation Memory• WITHREADONLYMEM_OPT include the read-only Translation Memory databases• WITHDOCMEM_OPT include the Translation Memory databases specified in the document properties• OVERWRITE_OPT overwrite any existing exported folder• MASTERFOLDER_OPT export as master folder• XLIFF_OPT export as XLIFF folder</div>
pszDocuments	pointer to the buffer containing the list of documents or NULL to export all documents of the folder
pszDescription	pointer to the buffer containing the export description or NULL if not used

Returns:

0 if successful or an error code in case of failures

EfqExportMem

```
USHORT EqfExportMem ( HSESSION  hSession,  
  
                     PSZ        pszMemname,  
  
                     PSZ        pszOutFile,  
  
                     LONG       lOptions  
  
                     )
```

Export a Translation Memory.

Parameters:

- | | |
|-------------------|--|
| <i>hSession</i> | the session handle returned by the EqfStartSession call |
| <i>pszMemname</i> | name of the Translation Memory |
| <i>pszOutFile</i> | fully qualified name of the output file |
| | options for Translation Memory export |
| | <ul style="list-style-type: none">• OVERWRITE_OPT to overwrite existing files |
| | and one of the format options |
| <i>lOptions</i> | <ul style="list-style-type: none">• TMX_OPT import in TMX format• XLIFF_MT_OPT import in XLIFF format• UTF16_OPT export in the EXP format (Unicode UTF-16 encoded)• ANSI_OPT export in the EXP format (ANSI encoded)• ASCII_OPT export in the EXP format (ASCII encoded) |
| | for the TMX_UTF8_OPT and TMX_UTF16_OPT also the following option can be specified |
| | <ul style="list-style-type: none">• TMX_NOCRLF_OPT to remove line breaks within the segment data |

Returns:

0 if successful or an error code in case of failures

EqfExportSegs

```
USHORT EqfExportSegs ( HSESSION  hSession,
                       PSZ        pszFolderName,
                       PSZ        pszDocuments,
                       PSZ        pszStartStopFile,
                       PSZ        pszOutFile,
                       LONG        lOptions
                       )
```

Export segments contained in specific tag groups.

Parameters:

hSession	the session handle returned by the EqfStartSession call
pszFolderName	name of the folder
pszDocuments	list with document names or NULL to process all documents of the folder
pszStartStopFile	file containing the start/stop tag list
pszOutFile	fully qualified name of the output file receiving the segment data
lOptions	options for the processing <ul style="list-style-type: none">OVERWRITE_OPT overwrite any existing output fileCOMPLETE_IN_ONE_CALL_OPT complete processing in one single call to the API

Returns:

0 if successful or an error code in case of failures

EfqFilterNoMatchFile

```
unsigned short EqfFilterNoMatchFile ( HSESSION    hSession,

                                     char *        pszInNoMatchXML,

                                     char *        pszGlobMemOptionFile,

                                     char *        pszMemory,

                                     char *        pszOutNoMatchXML,

                                     char *        pszOutNoMatchEXP,

                                     char *        pszWordCountReport,

                                     long          lOptions

                                     )
```

Checks matches from a NOMATCH file against a memory and applies any Global Memory option file.

This API function looks up all matches contained in a NOMATCH file (in XML format) in the given memory and applies the specified Global Memory option file on the memory proposals. The function creates a memory match word count and writes any matches not found in the input memory to a new NOMATCH file. The new NOMATCH file can be in the XML format and/or the .EXP format. The processing is done in small units and the API call is to be called repetitively as long as the return code CONTINUE_RC is returned. To do the processing in one block specify the option COMPLETE_IN_ONE_CALL_OPT. The word count report can be created in the XML format (use the option XML_OUTPUT_OPT) or in plain text format (use the option TEXT_OUTPUT_OPT). The word count report creation in plain text format is the default.

Parameters:

<i>hSession</i>	the session handle returned by the EqfStartSession call
<i>pszInNoMatchXML</i>	fully qualified file name of the input NOMATCH file in XML format
<i>pszGlobMemOptionFile</i>	fully qualified file name of the Global Memory option file
<i>pszMemory</i>	Name of the internal memory being used for the look-up.
<i>pszOutNoMatchXML</i>	fully qualified file name of the new NOMATCH file in the XML format (can be NULL when not used)
<i>pszOutNoMatchEXP</i>	fully qualified file name of the new NOMATCH file in the EXP format (can be NULL when not used)
<i>pszWordCountReport</i>	fully qualified file name of the created memory match word count report (can be NULL when not used)
	options for the processing
<i>lOptions</i>	<ul style="list-style-type: none">COMPLETE_IN_ONE_CALL_OPT to do the processing in one call (rather than doing the processing in small units)TEXT_OUTPUT_OPT to create the word count report in plain text format (=default)XML_OUTPUT_OPT to create the word count report in XML format

Returns:

0 if successful or an error code in case of failures

EqfFolderExists

```
USHORT EqfFolderExists ( HSESSION  hSession,
                          PSZ        pszFolderName
                          )
```

Check the existence of a folder.

Parameters:

- hSession the session handle returned by the EqfStartSession call
- pszFolderName name of the folder

Returns:

0 when the folder exists or an error code

EqfFreeSegFile

USHORT EqfFreeSegFile (HPARSSEGFILE *hLoadedFile*)

Free all memory occupied by a loaded file.

Parameters:

hLoadedFile the loaded file handle returned by the EqfLoadSegFile call

Returns:

0 if successful or an error code in case of failures

EqfGetFolderProp

```
USHORT EqfGetFolderProp ( HSESSION      hSession,
                          PSZ            pszFolderName,
                          PEXTFOLPROP    pExtFolProp
                          )
```

Get the properties of a folder.

Parameters:

- hSession the session handle returned by the EqfStartSession call
- pszFolderName name of the folder to get the properties from
- pExtFolProp pointer to a EXTFOLPROP buffer receiving the properties of the folder on return

Returns:

0 if successful or an error code in case of failures

EqfGetLastError

```
USHORT EqfGetLastError ( HSESSION  hSession,  
  
                        PUSHORT  pusRC,  
  
                        PSZ      pszMsgBuffer,  
  
                        USHORT    usBufSize  
  
                        )
```

Get information about the last occurred error.

Parameters:

- hSession* the session handle returned by the EqfStartSession call
- pusRC* pointer to the buffer for last return code
- pszMsgBuffer* pointer to a buffer receiving the error message text
- usBufSize* size of message buffer in number of bytes

Returns:

0 if successful or an error code in case of failures

EqfGetMatchLevel

```
USHORT EqfGetMatchLevel ( HSESSION      hSession,
                          PEQFSEGINFO    pSegment,
                          PEQFSEGINFO    pProposal,
                          SHORT *         psMatchLevel,
                          SHORT *         psMatchState,
                          LONG            lOptions
                        )
```

Create a new folder.

Parameters:

hSession	the session handle returned by the EqfStartSession call
pSegment	pointer to the segment info
pProposal	pointer to the memory proposal info
psMatchLevel	pointer to caller's match level field
psMatchState	pointer to caller's match state field
lOptions	options to be used by the API call <ul style="list-style-type: none">NO_GENERIC_INLINETAG_REPL_OPT suppress inline tag replacementUSE_GENERIC_INLINETAG_REPL_OPT perform generic inline tag replacement

Returns:

0 if successful or an error code in case of failures

EqfGetProgress

```
USHORT EqfGetProgress ( HSESSION  hSession,  
  
                        PUSHORT  pusProgress  
  
                        )
```

Get the progress of currently running process.

Parameters:

hSession the session handle returned by the EqfStartSession call

pusProgress pointer to a buffer receiving the current progress value

Returns:

0 if successful or an error code in case of failures

EqfGetSegmentNumber

```
USHORT EqfGetSegmentNumber ( HPARSSEGFILE hLoadedFile,
                              LONG lLine,
                              LONG lColumn,
                              PLONG plSegNum
                              )
```

Get the segment number for a given line/column position.

Parameters:

- hSession* the session handle returned by the EqfStartSession call
- lLine* line position of segment
- lColumn* columns position of segment
- plSegNum* number of segment found at given position

Returns:

0 if successful or an error code in case of failures

EqfGetSegNum

```
USHORT EqfGetSegNum ( HPARSSEGFILE hLoadedFile,  
  
                     PLONG plNumOfSegs  
  
                     )
```

Get number of segments in loaded file.

Parameters:

- hLoadedFile* the loaded file handle returned by the EqfLoadSegFile call
- plNumOfSegs* pointer to a buffer receiving the number of segments in the loaded file

Returns:

0 if successful or an error code in case of failures

EqfGetSegW

```
USHORT EqfGetSegW ( HPARSSEGFILE    hSegFile,  
                   LONG              lSegNum,  
                   PPARSSEGMENTW    pSeg  
                   )
```

Get the data of the specified segment.

Parameters:

- hLoadedFile* the loaded file handle returned by the EqfLoadSegFile call
- lSegNum* number of the segment
- pSeg* pointer to a PARSSEGMENTW structure receiving the segment data

Returns:

0 if successful or an error code in case of failures

EqfGetShortName

```
USHORT EqfGetShortName ( HSESSION  hSession,  
  
                        USHORT      ObjectType,  
  
                        PSZ         pszLongName,  
  
                        PSZ         pszShortName  
  
                        )
```

Get the internally used short name for a folder, dictionary, Translation Memory, or document Attention: this API function will only work for the older OpenTM2 plugins. Newer plugins will (hopefully) not use short names anymore.

Parameters:

<i>hSession</i>	the session handle returned by the EqfStartSession call
	type of the object being processed
<i>ObjectType</i>	<ul style="list-style-type: none">FOLDER_OBJ object is a folderMEMORY_OBJ object is a Translation MemoryDICT_OBJ object is a dictionaryDOCUMENT_OBJ object is a document
<i>pszLongName</i>	long name of the object for documents also the folder name has to be specified in the form foldername:documentname
<i>pszShortName</i>	pointer to a buffer for the returned short name

Returns:

0 if successful or an error code in case of failures

EqfGetSourceLine

```
USHORT EqfGetSourceLine ( HPARSSEGFILE  hLoadedFile,
                           LONG           lSegNum,
                           PLONG          plStartLine,
                           PLONG          plEndLine
                           )
```

Get the line information for a specific segment.

Parameters:

- hSession* the session handle returned by the EqfStartSession call
- lSegNum* number of the requested segment
- plStartLine* points to a buffer for the segment start line
- plEndLine* points to a buffer for the segment end line

Returns:

0 if successful or an error code in case of failures

EqfGetSysLanguage

```
USHORT EqfGetSysLanguage ( HSESSION hSession,
                           PSZ pSystemPropLang
                           )
```

Get the default target language (aka system language)

Parameters:

- `hSession` the session handle returned by the EqfStartSession call
- `pSystemPropLang` buffer receiving the actual system language (OpenTM2 language name)

Returns:

0 if successful or an error code in case of failures

EzfGetVersionEx

```
USHORT EzfGetVersionEx ( PSZ pszVersion,
                          int length
                          )
```

Get OpenTM2 version information.

Parameters:

- pszVersion pointer to a buffer for the version string
- length size of the buffer for the version string

Returns:

version information

EqfImportDict

```
USHORT EqfImportDict ( HSESSION  hSession,

                        PSZ        pszInFile,

                        PSZ        pszDictName,

                        PSZ        pszPassword,

                        LONG        lOptions

                        )
```

Import a dictionary.

Parameters:

hSession	the session handle returned by the EqfStartSession call
pszInFile	fully qualified name of input file
pszDictName	name of the dictionary
pszPassword	password of the dictionary or NULL if not used
lOptions	dictionary import options one of the format options <ul style="list-style-type: none">ASCII_OPT import from SGML (ASCII encoded)ANSI_OPT import from SGML (ANSI encoded)UTF16_OPT import from SGML (Unicode UTF-16 encoded)DXT_UTF8_OPT import from DXT XML (UTF-8 encoded) and one of the handling options <ul style="list-style-type: none">COMBINE_OPT combine existing entries and imported entriesREPLACE_OPT replace existing entries with the and imported entriesIGNORE_OPT ignore imported entries when the entry already exists

Returns:

0 if successful or an error code in case of failures

EqfImportDoc

```
USHORT EqfImportDoc ( HSESSION  hSession,

                      PSZ        pszFolderName,

                      PSZ        pszFiles,

                      PSZ        pszMemname,

                      PSZ        pszMarkup,

                      PSZ        pszEditor,

                      PSZ        pszSourceLanguage,

                      PSZ        pszTargetLanguage,

                      PSZ        pszAlias,

                      PSZ        pszStartPath,

                      PSZ        pszConversion,

                      LONG        lOptions

                      )
```

Import a document.

Parameters:

<i>hSession</i>	the session handle returned by the EqfStartSession call
<i>pszFolderName</i>	name of folder receiving the documents
<i>pszFiles</i>	list of input files (documents) in form of a comma separated list enclosed in parentheses
<i>pszMemname</i>	document Translation Memory or NULL
<i>pszMarkup</i>	document markup or NULL
<i>pszEditor</i>	document editor or NULL
<i>pszSourceLanguage</i>	document source language or NULL
<i>pszTargetLanguage</i>	document target language or NULL
<i>pszAlias</i>	alias for document name or NULL
<i>pszStartPath</i>	optional start path

pszConversion document conversion - not used anymore ans should be NULL

lOptions document import options or 0

- OVERWRITE_OPT to overwrite any existing document with the imported one

Returns:

0 if successful or an error code in case of failures

EqfImportFolder

```
USHORT EqfImportFolder ( HSESSION  hSession,  
  
                          PSZ        pszFolderName,  
  
                          CHAR        chFromDrive,  
  
                          CHAR        chToDrive,  
  
                          LONG        lOptions  
  
                          )
```

Import a folder from the directory.

Parameters:

<i>hSession</i>	the session handle returned by the EqfStartSession call
<i>pszFolderName</i>	name of the folder being imported
<i>chFromDrive</i>	drive containing the imported folder
<i>chToDrive</i>	target drive for imported folder
	folder import options
<i>lOptions</i>	<ul style="list-style-type: none">• XLIFF_OPT import a XLIFF folder (when not specified the import of a standard FXP folder is assumed)• NOMARKUP_UPDATE_OPT do not import any markup tables contained in the exported folder• WITHDICT_OPT import any dictionaries contained in the exported folder• WITHMEM_OPT import any Translation Memory databases contained in the exported folder

Returns:

0 if successful or an error code in case of failures

EqfImportFolderFP

```
USHORT EqfImportFolderFP ( HSESSION    hSession,  
  
                           PSZ          pszFolderName,  
  
                           PSZ          pszFromPath,  
  
                           CHAR         chToDrive,  
  
                           LONG         lOptions  
  
                           )
```

Import a folder from the given directory.

Parameters:

<i>hSession</i>	the session handle returned by the EqfStartSession call
<i>pszFolderName</i>	name of the folder being imported
<i>pszFromPath</i>	the fully qualified path name of the directory containing the imported folder
<i>chToDrive</i>	target drive for imported folder
	folder import options
<i>lOptions</i>	<ul style="list-style-type: none">• XLIFF_OPT import a XLIFF folder (when not specified the import of a standard FXP folder is assumed)• NOMARKUP_UPDATE_OPT do not import any markup tables contained in the exported folder• WITHDICT_OPT import any dictionaries contained in the exported folder• WITHMEM_OPT import any Translation Memory databases contained in the exported folder

Returns:

0 if successful or an error code in case of failures

EqfImportMem

```
USHORT EqfImportMem ( HSESSION  hSession,  
  
                     PSZ        pszMemname,  
  
                     PSZ        pszInFile,  
  
                     LONG       lOptions  
  
                     )
```

Import a Translation Memory.

Parameters:

<i>hSession</i>	the session handle returned by the EqfStartSession call
<i>pszMemname</i>	name of the Translation Memory
<i>pszInFile</i>	fully qualified name of the input file
	options for Translation Memory import one of the format options
	<ul style="list-style-type: none">• TMX_OPT import in TMX format• XLIFF_MT_OPT import in XLIFF format• UTF16_OPT import in the EXP format (Unicode UTF-16 encoded)• ANSI_OPT import in the EXP format (ANSI encoded)• ASCII_OPT import in the EXP format (ASCII encoded)
<i>lOptions</i>	one of the markup table handling functions
	<ul style="list-style-type: none">• CANCEL_UNKNOWN_MARKUP_OPT stop import when an unknown markup is detected• SKIP_UNKNOWN_MARKUP_OPT skip segments with unknown markup• GENERIC_UNKNOWN_MARKUP_OPT use the default markup table for segments with unknown markup
	additional options which can be used
	<ul style="list-style-type: none">• CLEANRTF_OPT to remove RTF inline tags (only in combination with TMX_OPT)• IGNORE_OPT ignore invalid segments and continue the import

Returns:

0 if successful or an error code in case of failures

EqfLoadSegFile

```
USHORT EqfLoadSegFile ( HSESSION      hSession,  
  
                        PSZ            pszFullDocName,  
  
                        HPARSSEGFILE * phLoadedFile  
  
                        )
```

Load a segmented document into memory.

Parameters:

- hSession* the session handle returned by the EqfStartSession call
- pszFullDocName* fully qualified name of segmented document
- hLoadedFile* pointer to buffer for handle of loaded file

Returns:

0 if successful or an error code in case of failures

```
USHORT EqfMemoryExists ( HSESSION      hSession,  
  
                        PSZ            pszMemoryName  
  
                        )
```

Check the existence of a Translation Memory.

Parameters:

- hSession* the session handle returned by the EqfStartSession call
- pszMemoryName* name of the Translation Memory

Returns:

0 when the Translation Memory exists or an error code

EqfOpenDoc

```
USHORT EqfOpenDoc ( HSESSION  hSession,  
  
                   PSZ        pszFolderName,  
  
                   PSZ        pszDocument,  
  
                   ULONG      ulSegNum,  
  
                   ULONG      ulLine  
  
                   )
```

Open a document in OpenTM2 at the given location.

Parameters:

<i>hSession</i>	the session handle returned by the EqfStartSession call
<i>pszFolderName</i>	name of the folder
<i>pszDocument</i>	name of the document
<i>ulSegNum</i>	segment number of the segment to be activated
<i>ulLine</i>	line to be activated (ulSegNum has to be 0 to use the line number)

Returns:

0 if successful or an error code in case of failures

EqfOpenDocEx

```
USHORT EqfOpenDocEx ( HSESSION  hSession,  
  
                     PSZ        pszFolderName,  
  
                     PSZ        pszDocument,  
  
                     ULONG      ulSegNum,  
  
                     ULONG      ulLine,  
  
                     wchar_t *  pszSearch  
  
                     )
```

Open a document in OpenTM2 at the given location (extented version)

Parameters:

<i>hSession</i>	the session handle returned by the EqfStartSession call
<i>pszFolderName</i>	name of the folder
<i>pszDocument</i>	name of the document
<i>ulSegNum</i>	segment number of the segment to be activated
<i>ulLine</i>	line to be activated (ulSegNum has to be 0 to use the line number)
<i>pszSearch</i>	UTF-16 encode search string (ulSegNum and ulLine have to be 0 to search for a string)

Returns:

0 if successful or an error code in case of failures

EqfOrganizeMem

```
USHORT EqfOrganizeMem ( HSESSION hSession,  
  
                        PSZ pszMemname  
  
                        )
```

Organize a Translation Memory.

Parameters:

- hSession* the session handle returned by the EqfStartSession call
- pszMemname* name of the Translation Memory

Returns:

0 if successful or an error code in case of failures

EqfProcessNomatch

```
USHORT EqfProcessNomatch ( HSESSION  hSession,

                           PSZ        pszNomatch,

                           PSZ        pszInMemory,

                           PSZ        pszOutMemory,

                           PSZ        pszMemMatchReportText,

                           PSZ        pszMemMatchReportXml,

                           PSZ        pszDupReportText,

                           PSZ        pszDupReportXml,

                           LONG       lOptions

                           )
```

Process a SNOMATCH file Run one or more SNOMATCH (Segments with NO MATCH in the searched Translation Memory databases) files against a Translation Memory and copy all relevant memory proposals to the output memory, for the segments contained in the SNOMATCH files a memory match count report and a duplicate word count report is created. When the option COMPLETE_IN_ONE_CALL_OPT The processing is performed in small units and the API has to be called repetively until a return code other than CONTINUE_RC uis returned.

Parameters:

hSession	the session handle returned by the EqfStartSession call
pszNomatch	fully qualified name of a single SNOMATCH file, the name of a folder containing SNOMATCH files or a search path (using wild-card characters) for a group of SNOMATCH files
pszInMemory	name of the internal input memory
pszOutMemory	name of internal output memory (is created if it does not exist)
pszMemMatchReportText	fully qualified file name of the memory match count report (text format)
pszMemMatchReportXml	fully qualified file name of the memory match count report (XML format)
pszDupReportText	fully qualified file name of the duplicate word count report (text format)
pszDupReportXml	fully qualified file name of the duplicate word count report (XML format)
lOptions	options for processing <ul style="list-style-type: none">COMPLETE_IN_ONE_CALL_OPT complete the processing in one call to the API

Returns:

0 if successful or an error code in case of failures

EqfProcessNomatchEx

```
USHORT EqfProcessNomatchEx ( HSESSION  hSession,

                             PSZ         pszNomatch,

                             PSZ         pszInMemory,

                             PSZ         pszOutMemory,

                             PSZ         pszMemMatchReportText,

                             PSZ         pszMemMatchReportXml,

                             PSZ         pszDupReportText,

                             PSZ         pszDupReportXml,

                             PSZ         pszOutNomatchXml,

                             PSZ         pszOutNomatchExp,

                             LONG         lOptions

                             )
```

Process a SNOMATCH file (version with additional parameters) Run one or more SNOMATCH (Segments with NO MATCH in the searched Translation Memory databases) files against a Translation Memory and copy all relevant memory proposals to the output memory, for the segments contained in the SNOMATCH files a memory match count report and a duplicate word count report is created. When the option COMPLETE_IN_ONE_CALL_OPT The processing is performed in small units and the API has to be called repetively until a return code other than CONTINUE_RC uis returned.

Parameters:

<i>hSession</i>	the session handle returned by the EqfStartSession call
<i>pszNomatch</i>	fully qualified name of a single SNOMATCH file, the name of a folder containing SNOMATCH files or a search path (using wild-card characters) for a group of SNOMATCH files
<i>pszInMemory</i>	name of the internal input memory
<i>pszOutMemory</i>	name of internal output memory (is created if it does not exist)
<i>pszMemMatchReportText</i>	fully qualified file name of the memory match count report (text format)
<i>pszMemMatchReportXml</i>	fully qualified file name of the memory match count report (XML format)
<i>pszDupReportText</i>	fully qualified file name of the duplicate word count report (text format)
<i>pszDupReportXml</i>	fully qualified file name of the duplicate word count report (XML format)
<i>pszOutNomatchXml</i>	fully qualified file name of the output nomatch file in nfluent XML format

pszOutNomatchExp	fully qualified file name of the output nomatch file in OpenTM2 EXP format
Options	options for processing
	<ul style="list-style-type: none">• COMPLETE_IN_ONE_CALL_OPT complete the processing in one call to the API

Returns:

0 if successful or an error code in case of failures

EqfReduceToStemForm

```
unsigned short EqfReduceToStemForm ( HSESSION  hSession,
                                     char *      pszLanguage,
                                     char *      pszInputTerms,
                                     char *      pszInputFile,
                                     char *      pszReport,
                                     LONG        lOptions
                                     )
```

Reduce a list of words to their stem form.

Parameters:

hSession	OpenTM2 session handle returned by EqfStartSession
pszLanguage	name of the language being used for the spell checking
pszInputTerms	a comma separated list of terms or NULL if a input file is being used
pszInputFile	the fully qualified name of a plain text file containing the terms, one term per line or NULL if pszInputTerms is being used
pszReport	name of the report file receiving the results of the operation
lOption	options for the output of the report <ul style="list-style-type: none">TEXT_OUTPUT_OPT for plain text output (CSV) orXML_OUTPUT_OPT (= default) for XML output

Returns:

0 if successful or an error code

EqfRemoveDocs

```
USHORT EqfRemoveDocs ( HSESSION  hSession,
                        PSZ        pszFolderName,
                        PSZ        pszListFile
                      )
```

Remove a group of documents based on a list of document names in a plain text file.

Parameters:

- hSession the session handle returned by the EqfStartSession call
- pszFolderName name of the folder containing the documents
- pszListFile fully qualified name of text file containing the list of document names

Returns:

0 if successful or an error code in case of failures

EqfRename

```
USHORT EqfRename ( HSESSION  hSession,

                   USHORT      usMode,

                   PSZ         pszName,

                   PSZ         pszNewName,

                   LONG         lOptions

                   )
```

Rename a folder, Translation Memory, or a dictionary.

Parameters:

hSession	the session handle returned by the EqfStartSession call
usMode	rename mode <ul style="list-style-type: none">• RENAME_FOLDER to rename a folder• RENAME_MEMORY to rename a Translation Memory• RENAME_DICTIONARY to rename a dictionary
pszName	name of the object being renamed
pszNewName	new name of the object
lOptions	options for processing <ul style="list-style-type: none">• ADJUSTREFERENCES_OPT adjust any references to the renamed object

Returns:

0 if successful or an error code in case of failures

EqfRestoreDocs

```
USHORT EqfRestoreDocs ( HSESSION hSession,
                        PSZ pszFolderName
                        )
```

Restore a group of documents removed using the EqfRemoveDocs API call.

Parameters:

- hSession the session handle returned by the EqfStartSession call
- pszFolderName name of the folder containing the documents

Returns:

0 if successful or an error code in case of failures

EqfSetSysLanguage

```
USHORT EqfSetSysLanguage ( HSESSION  hSession,
                           PSZ         pSystemPropLang
                           )
```

Set the default target language (aka system language)

Parameters:

- hSession the session handle returned by the EqfStartSession call
- pSystemPropLang the new system language (OpenTM2 language name)

Returns:

0 if successful or an error code in case of failures

EqfSimpleGetmatchLevel

```
SHORT EqfSimpleGetMatchLevel ( HSESSION  hSession,

                               WCHAR *    pszSegmentText,

                               WCHAR *    pszProposalText,

                               PSZ        pszLanguage,

                               PSZ        pszSegmentMarkup,

                               PSZ        pszProposalMarkup,

                               LONG        lOptions

)
```

Retrieve the match level of a memory proposal (simplified calling syntax)

Parameters:

hSession	the session handle returned by the EqfStartSession call
pszSegmentText	segment text (Unicode UTF-16 encoded)
pszProposalText	memory proposal text (Unicode UTF-16 encoded)
pszLanguage	source language of segment and memory proposal (OpenTM2 language name)
pszSegmentMarkup	markup table name of the segment
pszProposalMarkup	markup table name of the memory proposal
lOptions	options to be used by the API call <ul style="list-style-type: none">NO_GENERIC_INLINETAG_REPL_OPT suppress inline tag replacementUSE_GENERIC_INLINETAG_REPL_OPT perform generic inline tag replacement

Returns:

0 if successful or an error code in case of failures

EcfStartSession

USHORT EcfStartSession (PHSESSION *phSession*)

Start a OpenTM2 API call session.

Parameters:

phSession pointer to a buffer receiving the session handle

Returns:

0 if successful or an error code in case of failures

EqfUpdateSegW

```
USHORT EqfUpdateSegW ( HPARSSEGFILE hSegFile,
                        LONG lSegNum,
                        PPARSSEGMENTW pSeg
                      )
```

Update the data of a specific segment.

Parameters:

- hLoadedFile** the loaded file handle returned by the EqfLoadSegFile call
- lSegNum** number of the segment
- pSeg** pointer to a PARSSEGMENTW structure containing the new segment data

Returns:

0 if successful or an error code in case of failures

EqfWriteSegFile

```
USHORT EqfWriteSegFile      ( HPARSSEGFILE  hLoadedFile,  
                             PSZ             pszFullDocName  
                             )
```

Create a new folder.

Parameters:

hLoadedFile the loaded file handle returned by the EqfLoadSegFile call

pszFullDocName fully qualified name of the segmented document file

Returns:

0 if successful or an error code in case of failures

