

OpenTM2
Translator's Reference

Version 0.03

[TABLE OF CONTENT]

Inhaltsverzeichnis

<i>About this book</i>	10
<i>Related information</i>	10
<i>Starting with OpenTM2</i>	11
<i>Before you start</i>	11
<i>Hardware requirements</i>	11
<i>Software requirements</i>	11
<i>Installing OpenTM2</i>	11
<i>Installation procedure</i>	12
<i>Installation step-by-step (NEW installation)</i>	12
<i>Installation step-by-step (UPGRADE installation)</i>	12
<i>Compatibility notes concerning Unicode support</i>	12
<i>Required settings</i>	12
<i>Starting OpenTM2</i>	12
<i>Getting help</i>	12
<i>Backing up your translation material</i>	13
<i>Defining additional drives</i>	13
<i>Working with the Translation Workbench</i>	13
<i>What the Workbench is</i>	13
<i>Arranging the windows</i>	14
<i>Defining the view details</i>	14
<i>Hiding and shrinking path names</i>	14
<i>Option to sort document list on file name and extension</i>	14
<i>Saving the Workbench</i>	15
<i>Working with the Translation Environment</i>	15
<i>What the Translation Environment is</i>	15
<i>What the colors mean</i>	16
<i>How markup tags are displayed</i>	16
<i>Changing the display of markup information</i>	16
<i>Displaying the original of a translation proposal</i>	17
<i>Making a window active</i>	18
<i>Checking how segments were translated</i>	19
<i>Influence of Windows regional settings (Windows control panel => "Regional and Language Options")</i>	19
<i>Working with bidirectional language documents</i>	19
<i>Prerequisites for bidirectional language support</i>	19
<i>Controlling the bidirectional display</i>	20
<i>Translation Environment with an Arabic document</i>	21
<i>Translation Environment with a Hebrew document</i>	23
<i>Further considerations</i>	23
<i>Working with Thai language documents</i>	24
<i>Preparing for a translation</i>	24
<i>Translation Environment with a Thai document</i>	25
<i>Using Find and Replace</i>	27
<i>Using Thai dictionaries and abbreviation lists</i>	27
<i>Translating HTML documents</i>	29
<i>Limitations of the current Thai support</i>	29
<i>Commenting segments in the editor</i>	30
<i>Working with the OpenTM2 editor</i>	31
<i>What the OpenTM2 editor is</i>	31
<i>Basic functions</i>	31
<i>Assigning keys</i>	31
<i>Finding and replacing terms</i>	31
<i>Searching for terms in Translation Memory databases</i>	32
<i>Moving around the text</i>	33
<i>Moving around marked text</i>	34
<i>Working with bookmarks</i>	35
<i>Working with lines and words</i>	35
<i>Translation-specific functions</i>	36
<i>Manipulating segments</i>	38
<i>Manipulating documents</i>	39
<i>Configuring the editor</i>	40
<i>Customizing colors and fonts</i>	40
<i>Displaying a ruler</i>	40
<i>Displaying the cursor position</i>	40
<i>Setting up the profile</i>	40
<i>Sharing a profile</i>	44
<i>Using OpenTM2 in the translation business</i>	44
<i>Translation scenarios</i>	44
<i>Translating a new document</i>	44
<i>Creating a folder</i>	45
<i>Importing a new document</i>	45
<i>Translating the document</i>	45

<i>Postediting and viewing the translated document</i>	45
<i>Exporting the translated document</i>	45
<i>Translating updated documents</i>	45
<i>Translating with one translator</i>	46
<i>Working with the Workbench</i>	46
<i>Working with the Translation Environment</i>	46
<i>Translating with several translators</i>	46
<i>Sharing translation resources</i>	46
<i>Exchanging data with other people and systems</i>	47
<i>Exchanging documents</i>	47
<i>Exchanging dictionaries</i>	47
<i>Exchanging terminology lists</i>	47
<i>Exchanging Translation Memory databases</i>	47
<i>Exchanging folders</i>	47
<i>Exchanging markup tables</i>	47
<i>Working with OpenTM2 components and functions</i>	47
<i>General concepts</i>	47
<i>How a task is described</i>	48
<i>Export and import</i>	48
<i>Printing</i>	48
<i>Printing list windows</i>	48
<i>Windows</i>	49
<i>Message windows</i>	49
<i>Progress windows</i>	49
<i>Menus</i>	50
<i>Specifying an output file</i>	50
<i>Results</i>	50
<i>Selecting multiple objects in list windows</i>	50
<i>Viewing and changing the details of a list item</i>	52
<i>Viewing selected details</i>	52
<i>Changing the details setting</i>	52
<i>Arranging and filtering list items</i>	53
<i>Arranging list items</i>	54
<i>Filtering list items</i>	55
<i>Viewing and changing the system preferences</i>	59
<i>Working with documents</i>	61
<i>Overview and terminology</i>	62
<i>What you can do with documents</i>	63
<i>Analyzing a document</i>	63
<i>Analyzing documents using defaults</i>	63
<i>Analyzing documents using Translation Memory databases</i>	64
<i>Editing analysis Profiles</i>	67
<i>Analyzing documents using dictionaries</i>	69
<i>Maintaining abbreviation lists</i>	71
<i>Closing a document</i>	72
<i>Counting the number of words in a document</i>	72
<i>Counting duplicate words in a document</i>	74
<i>Counting the number and type of matches in a document</i>	77
<i>Counting words in segments with fuzzy matches</i>	79
<i>Dealing with specific document formats</i>	81
<i>Ami Pro documents</i>	82
<i>ANSI documents</i>	82
<i>ASCII documents</i>	82
<i>Assembler documents</i>	82
<i>BookMaster (R) and Information Presentation Facility documents</i>	83
<i>FrameBuilder files</i>	83
<i>HTML documents</i>	83
<i>Interleaf files</i>	83
<i>Lotus (R) Notes (TM) documents</i>	83
<i>Microsoft (R) PowerPoint (R) documents</i>	83
<i>Microsoft (R) Word for Windows (R) documents</i>	83
<i>RTF documents</i>	83
<i>SGML documents</i>	83
<i>Unicode documents</i>	83
<i>Windows (R) help files</i>	84
<i>WordPerfect documents</i>	84
<i>Deleting a document</i>	84
<i>Exporting a document</i>	84
<i>Exporting a document in OpenTM2 format</i>	85
<i>Exporting a document in external format</i>	86
<i>Exporting a document in external format with a path</i>	87
<i>Exporting a document in validation format</i>	88
<i>Exporting a document in plain XML format</i>	91
<i>Exporting translated glossary terms to a Lotus (R) Notes (TM) database</i>	92
<i>Editing document revision marks for export</i>	93
<i>Importing a document</i>	94
<i>Importing a document in OpenTM2 format</i>	95
<i>Importing a document in external format</i>	96

Importing a document in external format with a path	98
Importing glossary terms from a Lotus (R) Notes (TM) database	100
Copying and moving documents between folders	101
Opening a document	101
Searching and replacing text in documents	102
Search segments with fuzzy matches	104
Spellchecking a document	106
Editing the spellcheck addendum	107
Translating a document	108
Opening and translating several documents	108
Viewing the properties of a document	109
Changing the properties of a document	110
Viewing the details of a document	112
Including notes for the translator.....	113
Translator's notes in a source document.....	113
Translator's notes in the Translation Environment	113
Preview XML documents	114
Step by step description to preview a XML document	114
Working with folders	115
Overview and terminology	115
What you can do with folders	116
Analyzing all the documents in a folder	116
Closing a folder	116
Counting the words in all documents of a folder	117
Creating a folder	117
Dictionary Selection window	119
Translation Memory Selection window.....	120
Creating a subfolder	121
Renaming a folder.....	122
Deleting a folder	122
Deleting a folder exported to the eqlexport subdirectory	123
Exporting a folder.....	124
Importing a folder	126
Importing a XLIFF folder	128
Importing a folder into controlled folder	129
Opening a folder	130
Searching and replacing text across several documents in a folder	130
Viewing the properties of a folder	131
Changing the properties of a folder	132
Viewing the details of a folder	134
Working with Translation Memory databases.....	135
Overview and terminology.....	135
Translation Memory matches	136
What you can do with Translation Memory databases.....	137
Creating a Translation Memory	137
Creating an external Translation Memory	138
Format of an external Translation Memory	139
The Translation Memory format.....	139
SGML tags for external Translation Memory databases	140
Control information of a Translation Memory segment	140
Renaming a Translation Memory	141
Deleting a Translation Memory.....	141
Exporting a Translation Memory	142
Creating an Initial Translation Memory.....	143
Importing a Translation Memory	146
Merging Translation Memory databases	147
Archiving a Translation Memory	148
Organizing a Translation Memory	149
Revising an Initial Translation Memory	149
The Initial Translation Memory editor	150
File menu	150
Edit menu	151
Align menu	151
Options menu	151
Cursor menu	152
Style menu	153
Revising a Translation Memory	153
Searching a translation	154
The Translation Memory editor	156
File menu	156
Edit menu	157
Options menu	159
Style menu	159
Revising external Translation Memory databases	159
Viewing the properties of a Translation Memory	159
Changing the properties of a Translation Memory	161
Viewing the details of a Translation Memory	162
Working with dictionaries	162

Dictionary entry structure	163
Dictionary search criteria	163
Overview and terminology	164
What you can do with dictionaries	164
Creating an SGML-based dictionary	165
The structure of an SGML-based dictionary	165
List of dictionary-entry fields and their SGML tags	168
Sample of an SGML-based dictionary	168
Creating a dictionary	169
Using an existing dictionary as model	172
Changing dictionary entry fields	172
Renaming a dictionary entry field	174
Adding a user-defined entry field	174
Defining a dictionary filter	175
Renaming a dictionary	177
Deleting a dictionary	178
Editing a dictionary entry	178
Exporting a dictionary	180
Importing a dictionary	181
Looking up a dictionary entry	182
Merging dictionaries	184
Opening a dictionary	185
Organizing a dictionary	185
Printing a dictionary	186
Defining the printout format	187
Protecting a dictionary	188
Searching for a dictionary entry	189
Viewing the properties of a dictionary	191
Changing the properties of a dictionary	192
Viewing the details of a dictionary	194
Option to select dictionary lookup for compound words in editor	194
Viewing style indicators for dictionary terms	195
Working with terminology lists	196
Overview and terminology	196
What you can do with terminology lists	196
Creating a terminology list	196
Creating a list of new terms or found terms	197
Creating an exclusion list	197
Creating an external terminology list	197
Deleting a terminology list	198
Exporting a terminology list	199
Importing a terminology list	200
Opening and changing a terminology list	201
Marking a term for a dictionary	203
Editing an exclusion list	204
Printing a terminology list	204
Viewing the details of a terminology list	204
Working with markup tables	205
Overview and terminology	206
What you can do with markup tables	206
Changing a markup table	207
Creating a markup table	207
Deleting a markup table	207
Exporting a markup table	208
Importing a markup table	209
Changing the properties of a markup table	210
Working with language-support files	215
Overview and terminology	215
What you can do with language-support files	215
Deleting language-support files	215
Extending language-support files	216
What you can do for other languages	216
Working with the samples	216
Overview and terminology	216
What you can do with the samples	217
Deleting a sample folder	217
Importing and opening a sample folder and its documents	217
Translation exercise with a sample document	218
Creating reports	218
Overview and terminology	218
Creating a History Report	221
Creating a Counting Report	223
Creating a Calculating Report	226
Creating a Preanalysis Report	230
Creating a Redundancy Report	233
Creating a Redundant Segment List	236
Changing the layout of a report	237
The report layout	237

<i>Changing the layout</i>	239
<i>Setting the factors for cost calculation</i>	242
<i>Working from the command area (EQFCMD)</i>	245
<i>How to read syntax diagrams</i>	246
<i>Analyzing a document or folder</i>	246
<i>Exporting documents</i>	248
<i>Importing documents</i>	250
<i>Deleting documents</i>	252
<i>Creating a folder</i>	253
<i>Exporting a folder</i>	254
<i>Importing a folder</i>	256
<i>Deleting a folder</i>	257
<i>Creating a Translation Memory</i>	257
<i>Deleting a Translation Memory</i>	258
<i>Exporting a Translation Memory</i>	259
<i>Importing a Translation Memory</i>	260
<i>Exporting a dictionary</i>	260
<i>Importing a dictionary</i>	261
<i>Archiving a Translation Memory</i>	262
<i>Examples</i>	263
<i>Organizing a Translation Memory</i>	263
<i>Opening a document</i>	263
<i>Counting words</i>	264
<i>Creating reports</i>	265
<i>Performing several tasks in one step</i>	267
<i>Working from the command area (EQFBATCH)</i>	268
<i>How to read syntax diagrams</i>	268
<i>Analyzing a document or folder</i>	269
<i>Archiving a Translation Memory</i>	270
<i>Creating reports</i>	271
<i>Counting words</i>	273
<i>Exporting a dictionary</i>	274
<i>Importing a dictionary</i>	275
<i>Deleting documents</i>	276
<i>Exporting documents</i>	277
<i>Importing documents</i>	279
<i>Creating a folder</i>	282
<i>Deleting a folder</i>	283
<i>Exporting a folder</i>	283
<i>Importing a folder</i>	285
<i>Creating a Translation Memory</i>	286
<i>Deleting a Translation Memory</i>	287
<i>Exporting a Translation Memory</i>	288
<i>Importing a Translation Memory</i>	289
<i>Opening a document</i>	290
<i>Organizing a Translation Memory</i>	290
<i>Renaming a folder, a dictionary or a Translation Memory</i>	291
<i>Performing several tasks in one step</i>	292
<i>Working from the command area (Other Tools)</i>	293
<i>How to read syntax diagrams</i>	293
<i>Converting a NLV memory to a source/source memory</i>	294
<i>Creating an Initial Translation Memory from the command line</i>	294
<i>Revising an Initial Translation Memory</i>	297
<i>Reversing a Translation Memory</i>	298
<i>Changing m-flagged segments</i>	298
<i>Specifying the quality of m-flagged segments</i>	299
<i>Changing the markup, target language and date of segments</i>	301
<i>Removing segments with identical source and target strings</i>	302
<i>Removing inline tagging from an external memory</i>	303
<i>Showing the contents of exported folders</i>	304
<i>Restoring vital Translation Manager property files</i>	304
<i>Checking the folder history data</i>	304
<i>Correcting the drive letter information of Translation Manager files</i>	305
<i>Converting TM SGML memory databases into TMX format</i>	305
<i>Converting external Memory databases from the TMX format into the EXP format</i>	306
<i>Changing the type of an exported folder</i>	307
<i>Programming interfaces</i>	307
<i>Application programming interface for adding editors</i>	307
<i>Data types</i>	308
<i>Return codes</i>	308
<i>API calls</i>	309
<i>EQFCLEAR</i>	309
<i>EQFCLOSE</i>	310
<i>EQFCONVERTFILENAMES</i>	310
<i>EQFDELSEG</i>	311
<i>EQFDICTLOOK</i>	311
<i>EQFFILECONVERSIONEX</i>	312
<i>EQFGETDICT</i>	313

<i>EQFGETDOCFORMAT</i>	314
<i>EQFGETPROP</i>	314
<i>EQFGETSEGNUM</i>	315
<i>EQFGETSOURCELANG</i>	315
<i>EQFGETTARGETLANG</i>	315
<i>EQFINIT</i>	316
<i>EQFQUERYEXITINFO</i>	316
<i>EQFSAVESEG</i>	317
<i>EQFSEGFILECONVERTASCII2UNICODE</i>	318
<i>EQFSEGFILECONVERTUNICODE2ASCII</i>	319
<i>EQFTRANSSEG</i>	319
<i>EQFWORDCNTPERSEG</i>	320
<i>EQFWRITEHISTLOG</i>	321
<i>Non-DDE application programming interface</i>	322
<i>Overview and terminology</i>	322
<i>Data types</i>	322
<i>Sample code</i>	323
<i>Calling interface reference</i>	323
<i>EqfAnalyzeDoc</i>	324
<i>EqfAnalyzeDocEx</i>	326
<i>EqfArchiveTM</i>	327
<i>EqfBuildSegDocName</i>	328
<i>EqfChangeFoIProps</i>	329
<i>EqfChangeFoIPropsEx</i>	330
<i>EqfChangeMFlag</i>	332
<i>EqfCleanMemory</i>	332
<i>EqfCountWords</i>	334
<i>EqfCreateCntReport</i>	335
<i>EqfCreateCountReport</i>	339
<i>EqfCreateControlledFolder</i>	340
<i>EqfCreateFolder</i>	342
<i>EqfCreateITM</i>	343
<i>EqfCreateMem</i>	346
<i>EqfCreateSubFolder</i>	347
<i>EqfDeleteDoc</i>	348
<i>EqfDeleteFolder</i>	349
<i>EqfDeleteMem</i>	349
<i>EqfEndSession</i>	350
<i>EqfExportDict</i>	350
<i>EqfExportDoc</i>	351
<i>EqfExportFolder</i>	352
<i>EqfExportFolderFP</i>	353
<i>EqfExportFolderFPAs</i>	355
<i>EqfExportMem</i>	356
<i>EqfFreeSegFile</i>	357
<i>EqfGetFolderProp</i>	358
<i>EqfGetLastError</i>	359
<i>EqfGetMatchLevel</i>	359
<i>EqfGetProgress</i>	361
<i>EqfGetSegNum</i>	362
<i>EqfGetSegW</i>	363
<i>EqfGetSegmentNumber</i>	364
<i>EqfGetSourceLine</i>	364
<i>EqfGetSysLanguage</i>	365
<i>EqfGetVersion</i>	366
<i>EqfImportDoc</i>	366
<i>EqfImportDict</i>	368
<i>EqfImportFolder</i>	368
<i>EqfImportFolderFP</i>	370
<i>EqfImportMem</i>	371
<i>EqfLoadSegFile</i>	372
<i>EqfOrganizeMem</i>	373
<i>EqfProcessNomatch</i>	373
<i>EqfRename</i>	375
<i>EqfSetSysLanguage</i>	376
<i>EqfStartSession</i>	377
<i>EqfUpdateSegW</i>	377
<i>EqfWriteSegFile</i>	378
<i>Working with external markup tables</i>	379
<i>Creating new markup tables</i>	379
<i>Layout and content of a markup table</i>	380
<i>Substitution characters in a markup table</i>	381
<i>SGML tags for markup table header</i>	381
<i>SGML tags for markup tags and markup attributes</i>	382
<i>Examples of markup data and corresponding markup tags</i>	383
<i>Creating user exits for markup tables</i>	385
<i>General user exit entry points</i>	386
<i>EQFPRESEG2</i>	386

EQFPRESEGEX	387
EQFPOSTSEGW	388
EQFPOSTSEGWEX	388
EQFPOSTTMW	389
EQFCHECKSEGW	389
EQFSHOW	390
EQFGETCURSEG	390
EQFGETCURSEGW	390
EQFGETNEXTSEG	391
EQFGETNEXTSEGW	391
EQFGETPREVSEG	392
EQFGETPREVSEGW	392
EQFBUILDDOCPATH	392
EQFGETINFO	393
EQFPREUNSEGW	394
EQFPOSTUNSEGW	395
EQFPOSTUNSEG2	395
<i>API calls for user exits</i>	396
EQFGETTAOPTIONS	396
EQFSETTAOPTIONS	396
EQFTAOPTIONS	396
<i>User exit entry points for context-dependent translations</i>	397
EQFGETCONTEXTINFO	397
EQFGETSEGCONTEXT	397
EQFUPDATECONTEXT	398
EQFCOMPARECONTEXT	398
<i>Parser application programming interface</i>	399
ParsInitialize	399
ParsBuildTempName	399
ParsLoadSegFile	400
ParsGetSegNum	400
ParsGetSeg	400
ParsGetSegW	401
ParsUpdateSeg	401
ParsUpdateSegW	401
ParsWriteSegFile	402
ParsMakeSGMLSegment	402
ParsMakeSGMLSegmentW	403
ParsConvert	403
ParsGetDocName	404
ParsGetDocLang	404
ParsSplitSeg	404
ParsSplitSegW	405
ParsFreeSegFile	405
ParsLoadMarkup	406
ParsTokenize	406
ParsTokenizeW	406
ParsGetNextToken	407
ParsFreeMarkup	408
ParsTerminate	408
<i>Appendixes</i>	408
<i>Overview of the OpenTM2 menus</i>	408
<i>Editor functions reference</i>	409
<i>Editor functions inside the Translation window</i>	409
<i>Cursor movement in window</i>	409
<i>Scrolling</i>	410
<i>Editing</i>	410
<i>Modes of editing</i>	411
<i>Changing the presentation of control tags</i>	411
<i>Document overall functions</i>	411
<i>Switching to other windows</i>	412
<i>Editor functions outside the Translation window</i>	412
<i>Using the clipboard</i>	412
<i>Switching to other windows</i>	412
<i>Functions in the Translation Memory window</i>	412
<i>Functions in the Dictionary window</i>	412
<i>Format files for printing a dictionary</i>	413
<i>Sample format files</i>	413
FORMAT1.FRM	413
FORMAT2.FRM	413
FORMAT3.FRM	414
FORMAT4.FRM	414
<i>Defining your own format file</i>	414
<i>Tags for defining the format of a dictionary printout</i>	416
<i>The <set...> attributes</i>	417
<i>The <repeat...> attributes</i>	417
<i>The <var...> attributes</i>	418
<i>Displaying markup information for Word documents</i>	419

<i>Markup data that can be changed</i>	419
<i>Markup data that must not be changed</i>	421
<i>Markup data outside translatable segments</i>	422
<i>Exchanging data with other OpenTM2 products</i>	423
<i>Untranslated segments file</i>	423
<i>Directory structure of OpenTM2</i>	423
<i>System limitations</i>	424
<i>Hints, tips, and technical notices</i>	425
<i>About Microsoft (R) Word documents</i>	425
<i>Disk space consumption with .doc files</i>	425
<i>Problem: Word files with embedded picture data cannot be exported</i>	425
<i>Automatic font conversion for translated RTF documents</i>	425
<i>Translating TOC and index sections in Word documents</i>	426
<i>EQFRTF/EQFMSWRD markup table and third-party program limitations</i>	426
<i>About bidirectional language processing</i>	427
<i>Windows (R) Help/RTF – Table columns not in reverse order</i>	427
<i>Problem determination</i>	427
<i>Accessibility</i>	427
<i>Colors, fonts and key assignment</i>	429
<i>Set colors</i>	430
<i>Set fonts</i>	430
<i>Assign keys</i>	430
<i>Keyboard shortcuts</i>	430
<i>Opening menus</i>	431
<i>Navigating in windows (Folder List window, Document List window, Translation Memory List window or Dictionary List window)</i>	431
<i>Default keyboard shortcuts in Translation window</i>	431
<i>Help</i>	431
<i>Special accessibility keyboard shortcuts</i>	431
<i>Menu bar choices</i>	431
<i>File Menu 1</i>	432
.....	432
<i>File Menu 2</i>	432
<i>View Menu</i>	432
<i>Utilities Menu</i>	432
<i>Window Menu</i>	433
<i>Help Menu</i>	433
<i>Edit Menu</i>	433
<i>Options Menu</i>	434
<i>Cursor Menu</i>	434
<i>Translation Menu</i>	434
<i>Spellcheck Menu</i>	435
<i>Style Menu</i>	435
<i>Trademarks</i>	436
<i>Glossary of terms and abbreviations</i>	436

About this book

This book is intended for users who work with OpenTM2 under Windows^(R). This book is for all users of OpenTM2 who are already familiar with the basic functions of this product. OpenTM2 basics are explained in *A Quick Tour* and the *Translator's Workbook*.

The *Translator's Reference* gives you information on the more advanced topics of translating with OpenTM2.

It provides you with:

- Comprehensive descriptions of all OpenTM2 components and their functions essential for doing your daily translation business
- Application programming interfaces
- Appendixes with detailed technical information.

An easy way to find information about a specific item is to look it up in the index. However, if you are not sure about the precise naming of a function, you can look through the table of contents to find a topic this function may belong to.

All changes to the previous edition are marked with a vertical bar.

Related information

OpenTM2 for Windows^(R): A Quick Tour. It teaches you the basics of translating with **OpenTM2**.
OpenTM2 for Windows^(R): Translator's Workbook. It helps you learn to use **OpenTM2**.

Starting with OpenTM2

Before you start

This chapter informs you about the prerequisites you need before you start working with OpenTM2.

Hardware requirements

To install OpenTM2 you need the following minimum hardware configuration:

1. Standard PC with 32bit processor.
2. Minimum 1GB RAM.
3. Minimum 1GB free disk space.

Software requirements

OpenTM2 Version 1.0 can run under any of the following operating systems:

To install OpenTM2 you need the following minimum software configuration:

1. Windows NT
2. Windows 2000
3. Windows XP
4. WINDOWS 7

To use OpenTM2 in a LAN (Local Area Network) environment, you need the appropriate LAN software.

Installing OpenTM2

Before you start, make sure that code page 850 is set on your computer. OpenTM2 requires this code page to perform all linguistic functions correctly. Refer to the documentation of your operating system to see how code page 850 is set.

Note that some target languages require specific code pages and operating systems:

Target language	Prerequisite
Arabic	Code page 864, country code 785
Chinese, simplified	Code page 936
Chinese, traditional	Code page 950
Greek	Code page 869
Hebrew	Code page 862, country code 972
Japanese	Country code 081, one of the following operating systems: <ul style="list-style-type: none">• Windows^(R) 7• Windows^(R) XP• Windows^(R) 2000 Japanese• Windows^(R) NT Japanese
Russian	Code page 866

Installation procedure

Installation step-by-step (NEW installation)

1. Download the actual OpenTM2 package (*.ZIP) to your local hard disk drive.
2. Unzip the content of the ZIP-file to your local hard disk drive (you'll find a root-directory named \OTM\ on your hard disk drive).
3. COPY the entire structure (incl. all sub-directories) of the OTM-directory to your disk's root-directory (e. g. c:\).
- IMPORTANT: do NOT install OTM into e. g. c:\Program Files\ . This does not work in the current version of OpenTM2.
4. Navigate to e. g. c:\OTM\WIN\ using e. g. the WINDOWS EXPLORER.
5. Run the OpenTM2.exe
6. Panel "System Preferences" opens and message "EQF0368: Select the default target language" appears.
7. Click "OK".
8. From "Default Target Language" select your TARGET language (e. g. GERMAN(REFORM)).
9. Still being in panel "System Preferences":
 - check "Use Explorer-like tree view" (an improved FOLDER LIST WINDOWS starts after re-starting OpenTM2),
 - set the "Logo display time in ms" to "0" (the start-up panel no longer appears).
10. Restart OpenTM2 (indicated by message "EQF0258: You have to restart OpenTM2 in order to activate the new mode for the folder and document list windows")
11. After having re-started OpenTM2, you are ready to work with OpenTM2 (the OpenTM2 workbench is displayed).

For more information on how to use OpenTM2, read the documentation:

- OpenTM2 Quick Guide
- OpenTM2 Reference Guide

Installation step-by-step (UPGRADE installation)

1. Download the actual OpenTM2 package (*.ZIP) to your local hard disk drive.
2. Unzip the content of the ZIP-file to your local hard disk drive (you'll find a root-directory named \OTM\ on your hard disk drive).
3. COPY the entire structure (incl. all sub-directories) of the OTM-directory to your disk's root-directory (e.g. c:\) and OVERRIDE the existing structure.
4. The UPDATE is completed.

Compatibility notes concerning Unicode support

OpenTM2 uses the Unicode UTF-16 format to store its internal data files, such as segmented documents, translation memories, dictionaries, terms lists, and others. This is transparent to the OpenTM2 user. In the Translation Environment you might notice that characters from different code pages can be simultaneously displayed. OpenTM2's STANDARD editor supports documents in UTF-16 format.

Required settings

Before translators start working with OpenTM2, the "Default Target Language" must be set. This is very important in order to properly handle the import and export of translation memories and dictionaries, e. g. when importing an external memory in the SGML ASCII or the SGML ANSI format the "Default target language" must exactly match the language of the target segments in the imported memory otherwise some special characters might be corrupted after the import. Click on "Set" to save the changes.

For the correct display of characters in the dictionary editor, translation memory editor, global find&replace; window, and the various list and dialog windows (e.g. folder list window) the Windows "**Regional options**" must match the target language of the documents and dictionaries being worked with. The Windows regional settings are changed in the Windows "**Control Panel**" under "**Regional Options**". Select the locale for the current target language in the first tab **and** the third tab of the "**Regional options**" window. Windows has to be restarted after this change.

Starting OpenTM2

To start OpenTM2, double-click the OpenTM2 icon on the desktop. After OpenTM2 is started, its main window (OpenTM2 workbench) is displayed. From here start all tasks required for translating with OpenTM2. Find a description of the basic tasks in the document *A Quick Tour*.

Getting help

OpenTM2 provides online help for each window. You can display these help texts by clicking **Help** in the respective window, or pressing F1. Read the help texts whenever you are not sure what to

do in a window.

OpenTM2 comes with *A Quick Tour* and a *Translator's Workbook*. These books teach you how to perform the basic tasks. If you work through both of them before you start working with OpenTM2, it will be easier for you to become familiar with this program.

A good way to learn about OpenTM2 is to take the following approach:

1. Read *A Quick Tour* to learn how to perform the first steps with OpenTM2.
2. Work through the *Translator's Workbook* to learn how to do all tasks necessary for a translation project.
3. Refer to the *Translator's Reference* if you want to know more about a certain function in OpenTM2.

Backing up your translation material

Make regular backup copies of your translation material, such as Translation Memories, dictionaries, and documents.

Use the **Export...** command on the **File** menu of the OpenTM2 main window to copy the respective files on any media.

Defining additional drives

OpenTM2 lets you define more drives in addition to the one where you installed the program (by selecting **Configure Drives...** from the **Utilities** menu). You can use this option when there is not much space left on the current drive. Files from this drive can then be transferred to the additional ones.

Working with the Translation Workbench

What the Workbench is

The **Workbench** is the first main window you see after you have started OpenTM2. With this window you start the actual translation and perform tasks closely related to translating, such as importing and exporting.

The Workbench is the environment for managing all OpenTM2 components, such as:

- Documents
- Folders
- Translation Memories
- Dictionaries

Each component is listed in an individual window. Usually you work with any of these components in the following way:

1. Select the item you want to work with from the respective window. For example, a document.
2. Select the task you want to perform with this item from the respective menu. For example, IMPORT.
3. Fill in the required information in the respective window. OpenTM2 needs this information to correctly perform the task you selected.
4. Press a button (for example, IMPORT) to start the specified task.

The most important tasks in **OpenTM2** are also represented as icons in the toolbar of the Workbench. When you move your cursor over the individual icons you get hover help, giving you a short description of what the icon represents.

A status bar at the bottom of the Workbench indicates your current action. For example, if you select **New...** from the **File** menu, the status bar displays "Create a new object". To display the status bar, point to **Toolbar** on the **View** menu and then select **Statusbar** from the submenu.

You find all OpenTM2 components listed in smaller, subsidiary windows on the Workbench. Sometimes these windows are minimized. You see them displayed as icons at the bottom of the screen. If you double-click on one of these icons, it is restored to its original size.

When starting OpenTM2 for the first time, the following windows are shown on the Workbench:

- The "Folder List" window shows the folders currently available in your system. Folders contain all translation material for one translation project. Read more about folders in the *Translator's Workbook*
- The "Translation Memory List" window (as icon) shows the Translation Memories currently available in your system. Translation Memories keep all sentences you ever translated, and provide translation proposals for previously translated text. Read more about Translation Memories in [Chapter x "Working with Translation Memory Databases" on page xx](#).
- The "Dictionary List" window shows the dictionaries currently available in your system. For the

translation of a document you can use more than one dictionary. Read more about working with dictionaries in [Chapter x “Working with Dictionaries” on page xx](#).

Each item in a list is preceded by a small icon indicating the type of item.

To work with other OpenTM2 components, such as language lists, terminology lists, or markup tables, open the respective window by selecting it from the **Utilities** menu.

Arranging the windows

You can position and size all subsidiary windows on the Workbench as you find them most convenient. When the Workbench is displayed for the first time, OpenTM2 has made a default arrangement for you.

The defaults are usually good choices for many translators, but you are free to rearrange and resize the windows into whatever position and size you like.

The window layout can best be determined by you according to your requirements and preferences. For example, have the windows positioned in a tiled way, overlaying each other, displayed as icons, or hidden behind other windows to reduce clutter on the screen.

The easiest way to do this is by placing the mouse pointer in the window's title bar and dragging the window into the new position. This is a basic function of the operating system you use. If you do not know how to reposition or resize windows, refer to the appropriate user documentation of your operating system.

Defining the view details

Usually you just see the names of documents, Translation Memories, dictionaries, and so on, listed in the respective windows. For example, in the "Document List" window OpenTM2 by default displays only the document names.

However, if you are interested in more details about a specific list item, this information can be easily retrieved.

You can view, for example:

- A description of a list item
- When a list item was last modified
- Where a list item is located
- The completion percentage of a translation

Because not every translator has the need for the same amount of detail, OpenTM2 lets you choose individually what you want to display.

You define this by selecting **Change details...** from the **View** menu. On the subsequent window **Change View Details** you select the details in which you are interested.

If, in the corresponding List windows, the columns displaying the individual details are too narrow or too wide, you can change them by dragging the borders of the column headers using the mouse.

Hiding and shrinking path names

Document names in the "Document List" window can be shown in various ways:

- Without the path name, for example, as `Mydoc.htm`.
- With the path name, for example, as `ps303\final\source\mydoc.htm`.
- With the path name shrunk to eight characters, for example, as `ps303\f;...\\mydoc.htm`.

Which way is appropriate depends on your preferences and the amount of view details you already specified for the "Document List" window. To hide or shrink the path names, select **Hide Path** or **Shrink Path** from the **View** menu. To show the path names, select **Show Path** from the **View** menu.

If the Name column in the "Document List" window is too narrow or too wide, you can drag the column border accordingly.

Option to sort document list on file name and extension

There are two new display columns of the document list "Name without path" and "Extension". These new columns can be used as sort or filter criteria in the same way as the already existing document list columns.

The column "Name without path" shows the document name without the relative path part, if the

document name does not contain a relative path this column shows the same name as the standard "Name" column.

The column "Extension" shows the extension of the document name which is the part of the document name following the last dot. If the document name has no extension its column is empty.

Saving the Workbench

When you have tailored the Workbench layout, defined the view details, and arranged the column widths as appropriate, you can save these modifications.

When you leave OpenTM2, you are asked if you want to save the current Workbench. If you confirm this, you get the same window layout on the Workbench the next time you start OpenTM2.

This saves you from rearranging windows for each OpenTM2 session.

Working with the Translation Environment

What the Translation Environment is

When you selected a document for translation and opened it, OpenTM2 takes you to the Translation Environment. It is the environment where you:

- Type the translation
- Can view and copy translations for specific terms
- Can view and copy previously translated text segments
- Can decide whether you use the keyboard or the mouse for a task

To save you from paper-shuffling, OpenTM2 automatically looks up terms in dictionaries and retrieves previous translations. You see the findings on the screen, and you can copy them if they are appropriate for your current context.

To conveniently do this, the Translation Environment is divided into the following subwindows:

- The "Translation" window
Displays the text to be translated, broken into segments. The first segment ready for translation is shown with a yellow background.
All formatting information is displayed in red. To avoid that it takes up too much space on your screen or distracts you from the actual text, you can reduce or hide it. By default it is protected from being overwritten. This ensures that the layout of the original text is retained in the translation.
In this window you perform the actual translation with an editor that provides functions that are especially useful for translating.
You do the actual translation by inserting or overtyping text. While editing the translation document, the original document is in the background. You can switch to it whenever necessary.
- The "Translation Memory" window
Displays translation proposals that are based on previously translated segments. OpenTM2 finds these proposals by looking through a database called Translation Memory that contains all text segments you ever translated together with their corresponding translations. When first created, the Translation Memory is empty. It is gradually filled as you continue translating. If it contains a segment that matches your currently active segment, OpenTM2 lets you know how you translated it before.
These proposals can be exact or fuzzy, depending on how precisely they match the current original segment. It is up to you to decide if you copy the proposal as is, or modify it to suit your current context.
Exact proposals can occur multiple times in a Translation Memory if the same sentence has been translated in different contexts. This fact is shown in the "Translation Memory" window by a [x++] string preceding the original segment, where x represents the number of available proposals. To display all proposals, double-click ++ or press the assigned shortcut key. To permanently set this option, see "[Setting up the profile](#)" on page xx.
If only the tagging differs between a segment and a proposal, the proposal is still considered to be an exact match. When you copy the proposal, its tagging is adapted to the tagging in the segment currently being translated.
If the Translation Memory offers more than one fuzzy proposal, the first proposal is most likely to fit in the current context. In the second or third fuzzy proposal the differences to the current original segment increase, and you have to be careful when you copy them.
To ease your decision which multiple fuzzy proposals to consider, you can customize OpenTM2 to display a quality number with every fuzzy proposal. This quality number (a percentage number) ranks different proposals, and you might just ignore proposals below a certain level. See "[Setting up the Profile](#)" on page xx about how to customize OpenTM2.
To learn how to copy a Translation Memory proposal see "[Translation-specific functions](#)" on

page xx.

- The "Dictionary" window

Displays translations that are found for terms in the currently active segment.

The title bar of this window contains the dictionaries used for the translation. Depending on what you specified on the Dictionary page of the "Customize Translation Functions" window (see "Setting up the profile" on page xx), the dictionaries and the translation of the source terms are numbered.

When a new segment is activated, OpenTM2 automatically looks up specific terms of this segment in the related dictionary. For example, if the term **proposals** is to be looked up, it is reduced to **proposal**. If a translation for **proposal** has been found in a dictionary, it is shown in the "Dictionary" window.

Whenever you activate a segment for translation, the "Translation Memory" and the "Dictionary" window reflect this by presenting appropriate translation proposals for this segment.

To learn how to copy a dictionary proposal see "Translation-specific functions" on page xx.

What the colors mean

OpenTM2 shows the status of segments in the "Translation" window by different colors. The default settings are:

Status	Text	Background	Meaning
Translated	Black	Grey	Segment has already been translated.
Active	Black	Yellow	Currently active segment ready to be translated.
Not translated	Blue	Grey	Segment has not been translated yet.
Not translatable	Red	Grey	Segment contains no translatable information.

How markup tags are displayed

OpenTM2 lets you choose how to display markup information in a translation document:

Style	Meaning
Protect	The markup tags are shown but you cannot change them (default).
Unprotect	The markup tags are shown and you can overwrite and change them.
Hide	The markup tags are not shown.
Shrink	All markup tags outside segments are not shown. A replacement character is shown instead.
Compact	The information that is not to be translated and the markup tags within a segment are replaced with a character.
Compact+1	Markup tags are shown up to a length of 10 characters and cannot be changed. Longer markup tags are followed by three dots (...), for example [Style=@Out...]

Changing the display of markup information

You can change the tag representation style by opening the **Style** menu and selecting a style command of your choice.

[Figure 1](#) contains examples of a Microsoft^(R) WORD document that show you how the text is presented, depending on the selected style. The following symbols are used to show information that is protected and cannot be translated:

- <>

Appears when you select shrink and compact style. The symbol is a placeholder for information that cannot be translated.

- #

Appears when you select compact style. This symbol shows that there is information in a

segment that cannot be translated.

These symbols are protected and have the same color as markup tags (the default is red). You can change them as follows:

1. Select **Profile Settings...** from the **Options** menu of the Translation Environment.
2. Select the **Display** page.
3. In the **Display Settings** box, change the respective symbols.

The shrink and compact styles are useful when you translate documents that contain numerous markup tags, such as RTF documents. The various style options let you display the text in a less cluttered form. The markup information is replaced by the symbols of your choice, and you can concentrate on the text that is to be translated.

If two or more markup tags appear after one another in a text, they are replaced by only one symbol depending on the style you chose.

Figure 2. Markup display styles

```
Protected/Unprotected:  
[style:heading 2]  
Planning Forms  
[paragraph: 0005,E8]  
[style:Normal]  
While reading this chapter, you will complete one or more of the  
following  
planning forms, which are included in [italic]Planning Forms[/italic],  
GX24-4093:  
[paragraph: 0005,F5]  
[style:Normal]  
[field: 3913]SYMBOL 183 \.f "Symbol" \.s 10 \h[field end: 0015] [efield]  
• Processor Configuration (Forms B3, B5, or B7)  
[paragraph: 0005,ED]  
[style:Normal]  
[field: 3913]SYMBOL 183 \.f "Symbol" \.s 10 \h[field end: 0015] [efield]  
• Integrated DASD/Tape Devices (Form C2).  
  
Hide:  
Planning Forms  
  
While reading this chapter, you will complete one or more  
of the following planning forms, which are included in  
Planning Forms, GX24-4093:  
• Processor Configuration (Forms B3, B5, or B7)  
• Integrated DASD/Tape Devices (Form C2).  
  
Shrink:  
<>  
Planning Forms  
<>  
While reading this chapter, you will complete one or more  
of the following planning forms, which are included in  
[italic]Planning Forms[/italic], GX24-4093:  
<>  
• Processor Configuration (Forms B3, B5, or B7)  
<>  
• Integrated DASD/Tape Devices (Form C2).  
  
Compact:  
<>  
Planning Forms  
<> While reading this chapter, you will complete one or more  
of the following planning forms, which are included in  
#Planning Forms#, GX24-4093:  
<>  
#Processor Configuration (Forms B3, B5, or B7)  
<>  
#Integrated DASD/Tape Devices (Form C2).
```

Displaying the original of a translation proposal

To view the source of a translation proposal double-click anywhere within the "Translation Memory" window. The "Source of Proposal(s)" window shows the original text of a proposal found in the Translation Memory, for example, the English original of a German translation. The original text of an exact match is preceded by [0], that of a fuzzy match by [f]. This window also shows the

segment that is currently active in the original document. The source of a proposal is displayed below the currently active segment. These segments are compared with one another, and differences are indicated by color.

You can change the colors by selecting **Colors...** from the **Options** menu of the "Translation" window.

OpenTM2 indicates the correspondence and differences between the source text of a found proposal and the currently active source text. This helps you determine how strongly you can rely on a translation proposal. If the source of the proposal differs from the segment currently to be translated, do not take the match as is, but change it so that the translation matches the original you are currently translating.

The following gives you an example how OpenTM2 indicates differences:

- Terms that occur in the source of the translation proposal but not in the currently active source segment.

Current segment:

0 - These are the descriptions of the products that will eventually be shown at the 2000 fair.

Translation proposal (German):

1[f] - Dies sind die Beschreibungen der Produkte, die dann auf der COMDEX Messe 2000 gezeigt werden.

Source of proposal:

1[f] - These are the descriptions of the products that will eventually be shown at the 2000 COMDEX fair.

The part that differs is underlined here. The proposal must be altered to reflect the meaning of the current segment.

- Terms that are different in the source of the match compared to the currently active segment.

Current segment:

0 - These are the descriptions of the products that will eventually be shown at the 2000 fair.

Proposal (German):

1[f] - Dies sind die Beschreibungen einiger Produkte, die dann auf der Messe 2000 gezeigt werden.

Source of proposal:

1[f] - These are the descriptions of **some** products that will eventually be shown at the 2000 fair.

The part that differs is shown in bold. The proposal must be changed to reflect the meaning of the current segment.

Terms that occur in the currently active segment but not in the source of the proposal.

Current segment:

0 - These are the descriptions of the products that will eventually be shown at the 2000 fair.

Proposal (German):

1[f] - Dies sind die Beschreibungen der Produkte, die auf der Messe 2000 gezeigt werden.

Source of proposal:

1[f] - These are the descriptions of the products that will ? be shown at the 2000 fair.

The part that differs is indicated by a question mark. The proposal must be altered to reflect the meaning of the current segment. That means, a translation for the term **eventually**, which is not included in the source of the proposal but in the current source, must be added at the ? position.

Text that is identical in the active segment and in the source of the proposal is not highlighted. However, you can have the identical parts displayed in a color of your choice.

Making a window active

The following windows are listed when you open the **File** menu in the Translation Environment:

- "Translation"
- "Translation Memory"
- "Dictionary"
- "Original"
- "Source of Proposal"

All documents you opened for reference purposes, in addition to the document you translate are listed. You open another document by selecting **Open...** from the **File** menu of the Translation Environment.

Use this option to quickly move to a certain window while working in the Translation Environment. It is especially helpful if a window is hidden behind another window. If you select this window from the list, it becomes immediately active and is displayed in the foreground.

Checking how segments were translated

Translated segments can have been translated from scratch, copied from a Translation Memory and changed, or copied from a Translation Memory and not changed.

OpenTM2 lets you know how many of the translated segments have been translated by one of these methods. When you save the translation, you can request a summary in the "Document List" window showing how many segments have been:

- Translated from scratch
- Copied from a Translation Memory proposal
- Copied from a Translation Memory proposal and then changed

This feature can be useful if translations are paid depending on the way they have been translated. Translated segments that have been translated from scratch may be paid higher than segments that have been copied from a proposal in the Translation Memory. It helps to find out how many segments have been translated in one of the above ways.

Influence of Windows regional settings (Windows control panel => "Regional and Language Options")

The regional settings of Windows control the display of textual data within standard entry fields and multi-line entry fields; i.e. when the regional settings have been set to "Chinese", German umlaut in description fields (folder description, memory description) will not be displayed correctly. As the Global Find&Replace function use a standard multi-line entry field for the found text, the displayed text may be incorrect when the regional settings do not match the target language of the documents being searched.

Note:

- The displayed text in the Translation Environment windows is not affected by the Windows regional settings as all data is in Unicode. The display may be incorrect however when the selected fonts do not support the characters being displayed.

Working with bidirectional language documents

Translation Environment bidirectional languages bidirectional languages Arabic languages see bidirectional languages Hebrew languages see bidirectional languages OpenTM2 enables you to translate documents from and to Arabic and Hebrew, and it supports Arabic and Hebrew dictionaries and spellchecking. These languages are "bidirectional" languages; the presentation of text is from right to left, and embedded numerals, or segments of text in languages such as English, are presented from left to right.

The following sections describe the prerequisites for the bidirectional language support in OpenTM2, and how you can control the display of bidirectional documents. The particularities of the Translation Environment and OpenTM2's built-in editor, when used with an Arabic and Hebrew document, are explained. Finally, minor items that you should be aware of when translating from or to a bidirectional language are listed.

Prerequisites for bidirectional language support

Under the Windows NT^(R), Windows^(R) 2000 and Windows XP operating systems OpenTM2 requires that the System Locale is set to Arabic or Hebrew in the system preferences. If you open an Arabic or Hebrew document and it does not display its text right-aligned and you cannot read it from right to left, you need to set the System Locale. Click **Start**, **Settings**, **Control Panel**, then select **Regional Settings**. You find a panel similar to the following. Set the System Locale and start over.

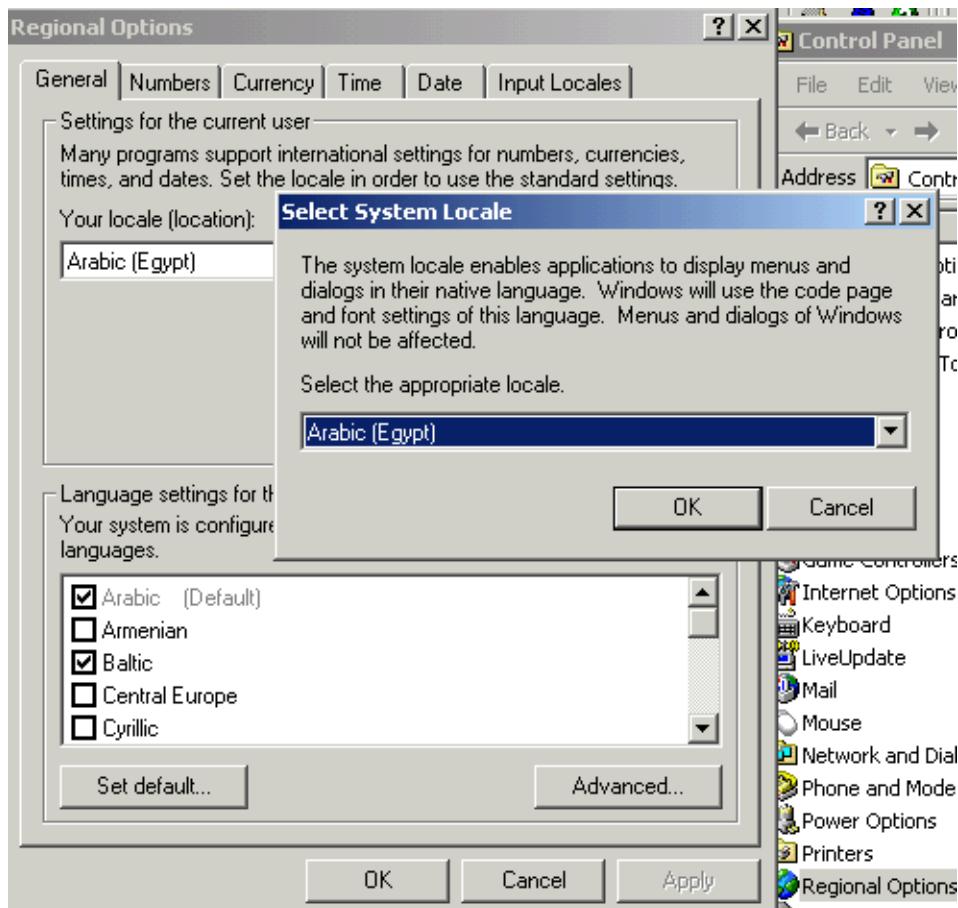


Figure 3. Setting the System Locale for bidirectional languages

If you do not have these settings correct, drop-down boxes and dialog control panels might not show text in the correct language.

Controlling the bidirectional display

You can control whether embedded text in a bidirectional language document, like numerals and foreign language text phrases, are displayed from left to the right or reverted.

The decision depends on your personal preference and on the amount of embedded text in a document.

customizing for bidirectional languages visual display option To change the direction, you can customize the Translation Environment as follows:

- Open the **Options** menu
- Select **Profile settings...**
- In the "Customize Translation Functions" window, select the **Display** page.
- Select or clear the **Visual display** check box, and click **Set** to save the profile settings.
- Selecting this check box displays embedded text in the same direction as the Arabic or Hebrew text, from right to the left.

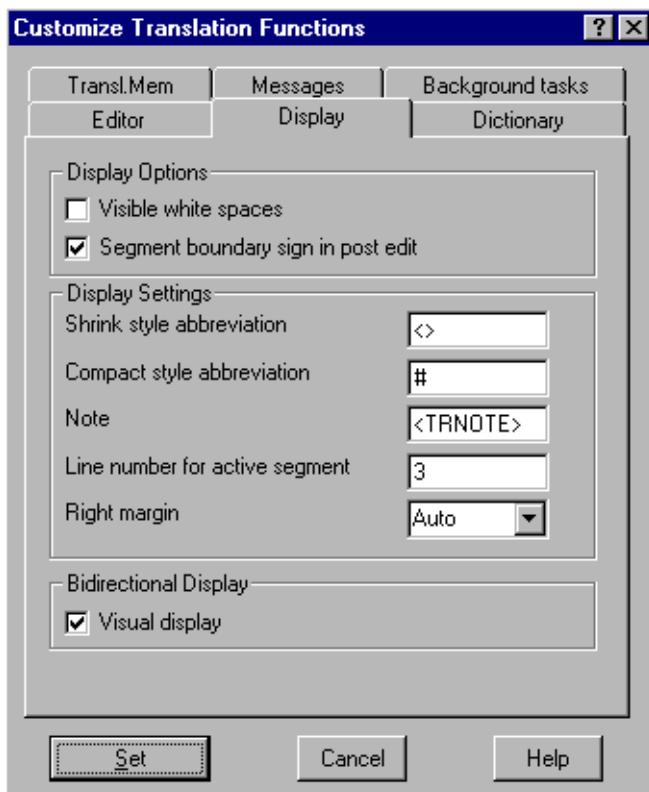


Figure 4. Controlling the bidirectional display

Remember to clear this check box if you work with non-bidirectional documents.

Translation Environment with an Arabic document

When you open an Arabic document, you recognize the usual "Translation" window, "Dictionary" window, and "Translation Memory" window. However, the contents of these windows look different in several aspects. The following Translation Environment example assumes a translation from English to Arabic:



Figure 5. Translation Environment with an Arabic document

text alignmentArabic language

reading orderArabic language

Arabic languagereading order text alignment

- In the "Translation" window the text is aligned to the right, and the ruler runs from right to the left (ruler digits are always shown as Latin digits). The reading order is from right to the left for English and Arabic words. While you are typing, the cursor moves from right to the left. The cursor control keys let the cursor move as depicted on the keys.
- In the "Dictionary" window all words are aligned to the left. The reading order of English words is from left to the right, and the reading order of Arabic words is from right to the left.
- In the "Translation Memory" window the English source text is aligned to the left, and its reading order is from left to the right. The Arabic translation proposals are aligned to the right, and their reading order is from right to the left (same alignment and reading order as in the "Translation" window).

Arabic languageshaping

shapingof Arabic characters

- Shaping of Arabic characters is supported. (Arabic characters can assume up to four different shapes depending on their positions in words and their connectivity traits of the surrounding characters. The proper shape is selected by a shape-determination routine.)

Arabic languageligating

ligatingof Arabic characters

- Ligatures of Arabic characters are supported. (A ligature is a graphic character consisting of two or more characters that are joined together. For example, the joining A and E form the ligature AE.)
- Digits in the proposal window are displayed as Latin digits.
- The **Find and Replace** function in the Translation Environment allows you to select the **Case Respect** or **Case Ignore** option. However, these options have no effect with Arabic language documents.

Translation Environment with a Hebrew document

When you open a Hebrew document, you recognize the usual "Translation" window, "Dictionary" window, and "Translation Memory" window. However, the contents of these windows look different in several aspects. The following Translation Environment example assumes a translation from English to Hebrew:

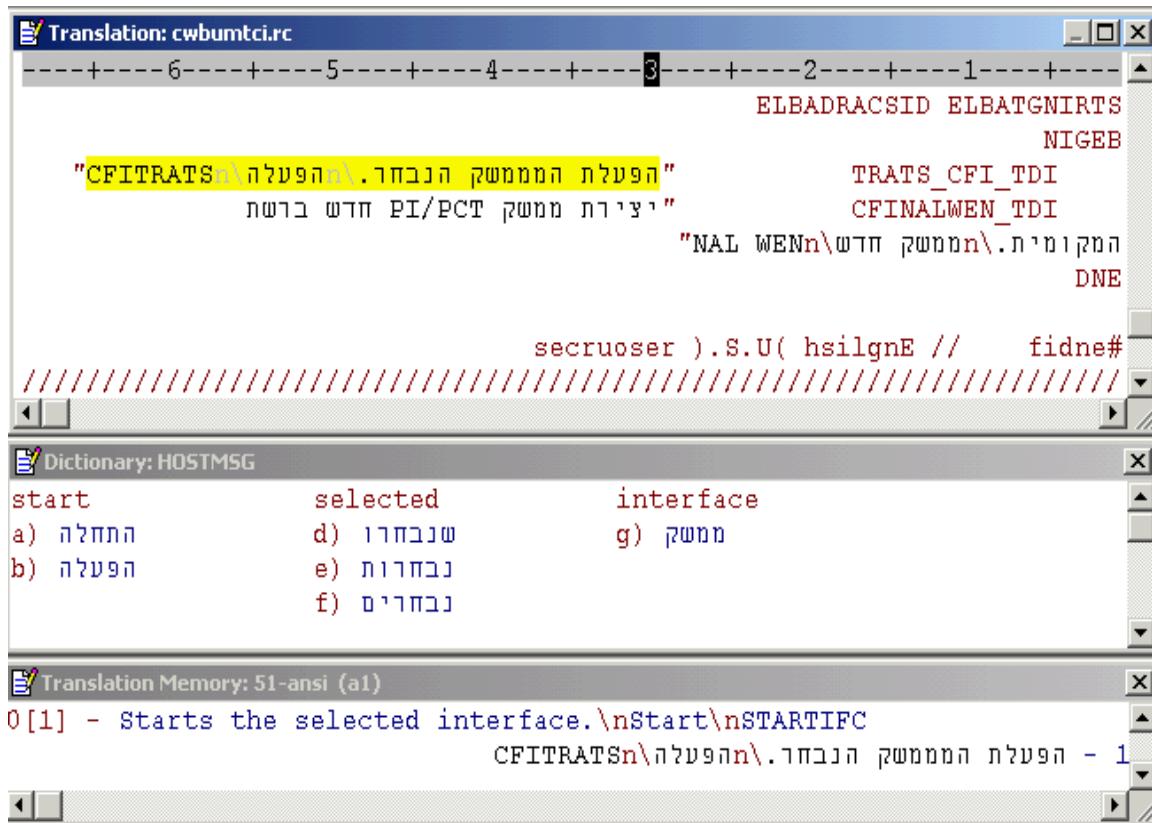


Figure 6. Translation Environment with a Hebrew document

text alignment
Hebrew language
reading order
Hebrew language
Hebrew language reading order text alignment

- In the "Translation" window the text is aligned to the right, and the ruler runs from right to the left (ruler digits are always shown as Latin digits). The reading order is from right to the left for English and Hebrew words. While you are typing, the cursor moves from right to the left. The cursor control keys let the cursor move as depicted on the keys.
- In the "Dictionary" window all words are aligned to the left. The reading order of English words is from left to the right, and the reading order of Hebrew words is from right to the left.
- In the "Translation Memory" window the English source text is aligned to the left, and its reading order is from left to the right. The Hebrew translation proposals are aligned to the right, and their reading order is from right to the left (same alignment and reading order as in the "Translation" window).

Further considerations

Defining shortcut keys

When a document with Arabic or Hebrew source or target language is opened, the keyboard properties are automatically changed to the Arabic, respectively Hebrew, input locale.

If you want to assign your own keys or key combinations for certain editor functions (**Options** menu, **Keys...**, **Assign Keys**), select the English input locale before doing so. This ensures a proper key assignment and a proper display of all key assignments. When finished, switch back to the previous input locale.

The treatment of braces - no symmetrical swapping

OpenTM2 does *not* interchange the order of some directional pairs of characters, such as left and right parentheses (), less-than and greater-than signs < >, left and right brackets [], and left and right braces { }.

< > is used in "Translation Memory" windows to denote matches. < > is used to denote shrink and compact styles.

You need to manually translate these characters in accordance with the logical sequence of a sentence.

The treatment of text in dialog boxes

In dialog boxes, for example, the Find and Replace box, the entry fields show the standard Windows^(R) behavior, regardless of whether the bidirectional language is active. Text is aligned to the left, and cursor movement is from left to the right. This behavior cannot be changed in OpenTM2. It is assumed to be acceptable because of the shortness of the text.

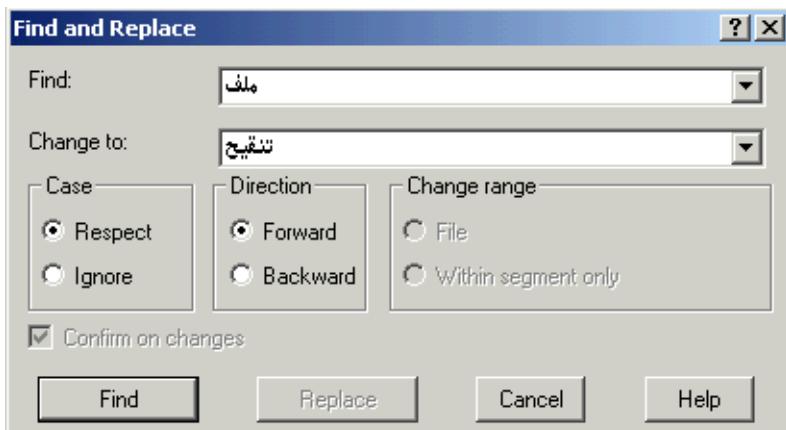


Figure 7. Dialog box example with a bidirectional language

Translation Memory considerations

When you edit a Translation Memory with the Translation Memory editor, Arabic and Hebrew source and target segments are aligned to the left, with a reading order from right to the left. During the creation of an Initial Translation Memory (with the **Create** option, which causes a visual presentation on the screen), the contents of Arabic and Hebrew documents are aligned to the left, with a reading order from right to the left.

Working with Thai language documents

OpenTM2 supports the translation of documents from and to the Thai language. This language is special in its character representation, character composition, word spacing, and writing order, therefore OpenTM2 provides a Rich Text Format (RTF) editor that can process Thai language documents.

This RTFEdit editor is required for Thai documents, but can also be used for all other document formats supported by OpenTM2. In contrast, OpenTM2's "Standard" editor supports all document formats except the Thai document format.

Another difference that you might notice: RTFEdit is a proportional fonts editor; the standard editor is a monospace fonts editor.

The following sections describe the preparation for a translation, shows a typical Translation Environment scenario, and lists particularities and limitations.

Preparing for a translation

When you start translating a Thai document for the first time, the Thai font is *not* automatically selected in the Translation Environment. You need to manually set the font.

From the Translation Environment, select **Fonts** from the **Options** menu. This takes you to the "Font" window:

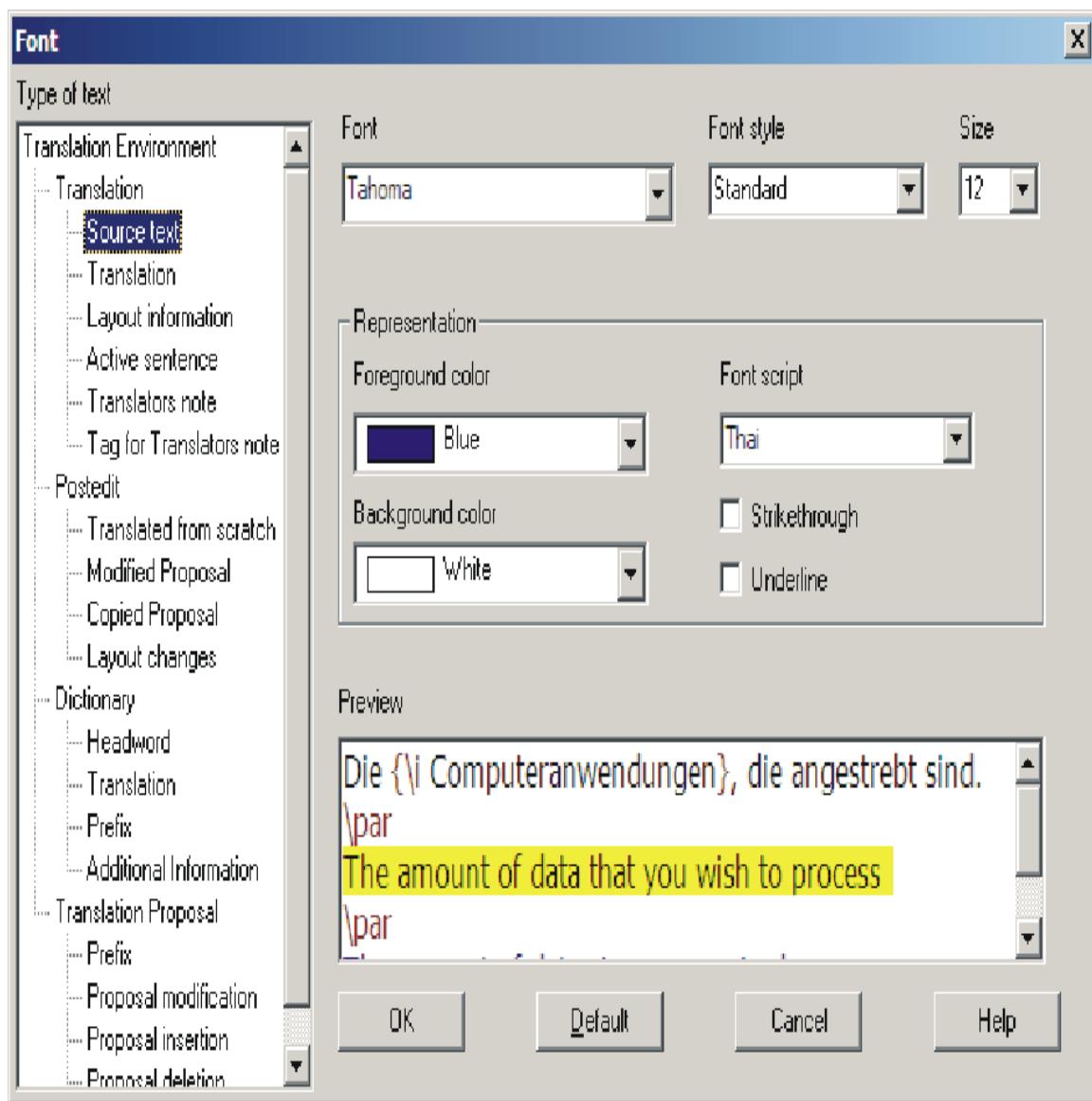


Figure 8. Setting fonts and colors for the Translation Environment

Select a font that supports the Thai script. You can also click **Defaults**, then **OK**, which selects an appropriate font for all Translation Environment windows shown on the left side of the "Font" window.

The window allows to customize and fine-tune the Translation Environment similar to OpenTM2's standard editor, as described in [Customizing colors and fonts](#). Be sure to select only fonts that support the Thai script. Your changes are reflected in the Preview area.

Translation Environment with a Thai document

When you open a Thai document, you see the usual three windows that make up the Translation Environment. In all windows Thai characters are properly displayed. You do not necessarily notice any differences because of the Rich Text Format editor, except that you can type and edit Thai characters.

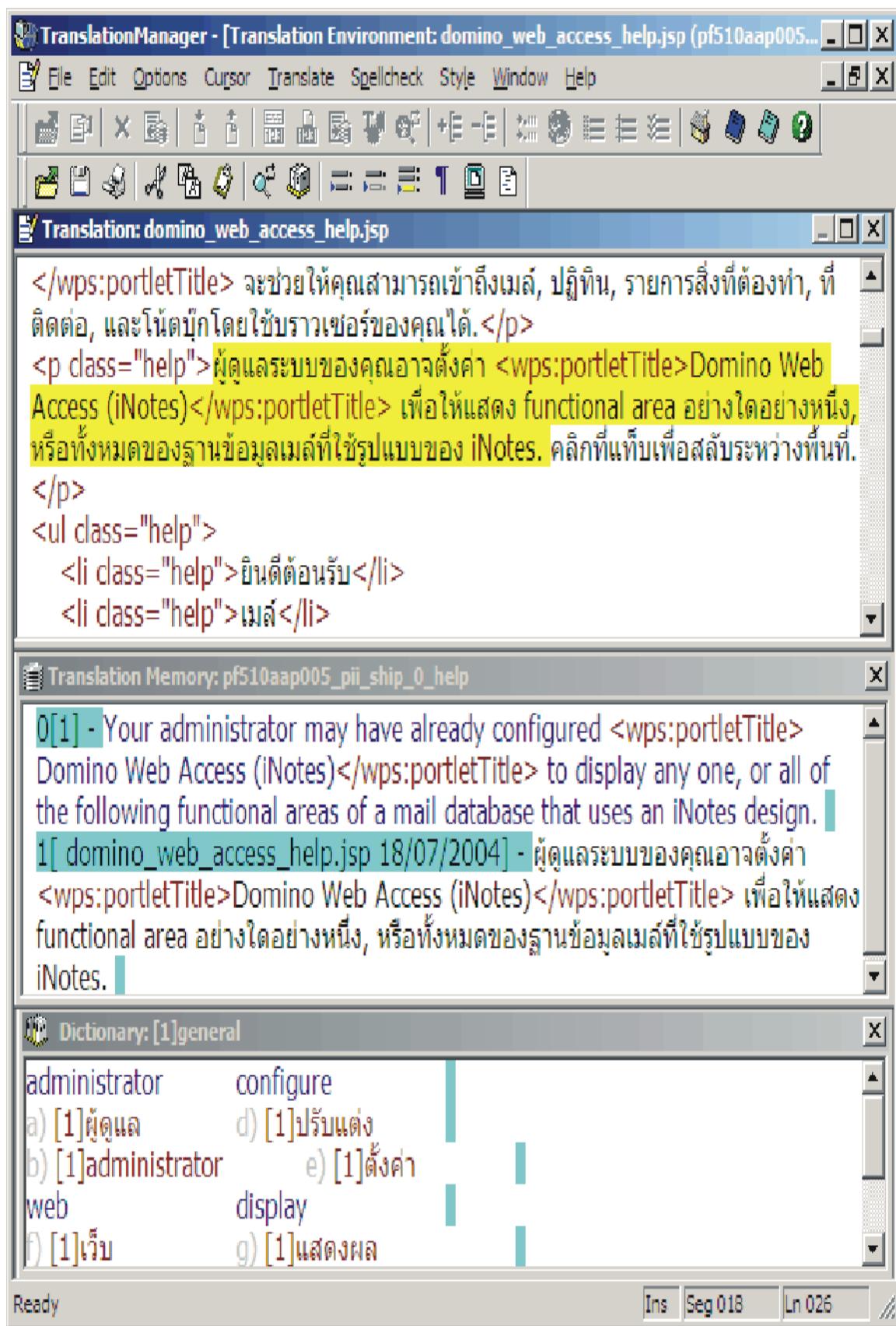


Figure 9. Translation Environment with a Thai document

The RTFEdit editor provides the same functions as the standard editor, with a few exceptions:

- Copying Translation Memory proposals with the `Ctrl-<number_of_proposal>` keys requires a non-Thai keyboard or the use of the numeric pad.
copying Translation Memory proposals
- When you open a document with a Thai source or target language with the RTFEdit editor, a Thai keyboard is automatically selected.

Spellchecking is available from the Translation Environment, as described in [Spellchecking a document](#). Note that no spelling aid is given for the Thai language.spellcheckingThai documents

Using Find and Replace

You can use the **Find and Replace** function in the Translation Environment with one exception: the **Case Respect** and **Case Ignore** options can be selected, but have no effect for the Thai language.

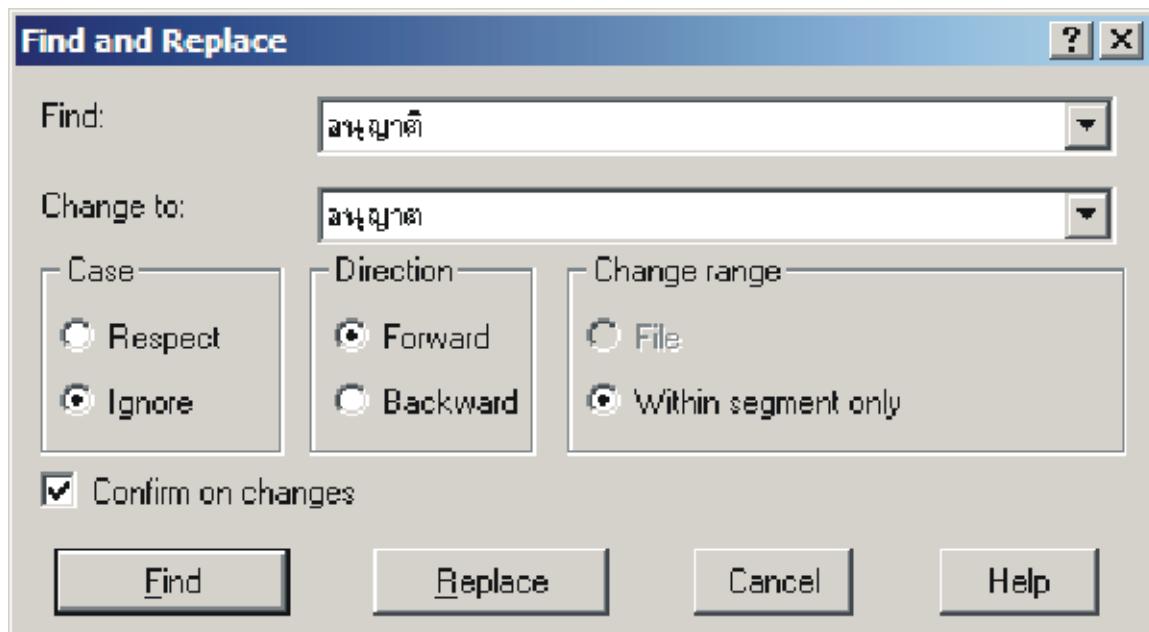


Figure 10. Using the **Find and Replace** window with Thai scripts

The **Global Find and Replace** function also works properly with Thai documents. When using this function with Thai text, ensure that the "Whole words only" option is not selected. The following example shows a search for a Thai character in Thai source files.

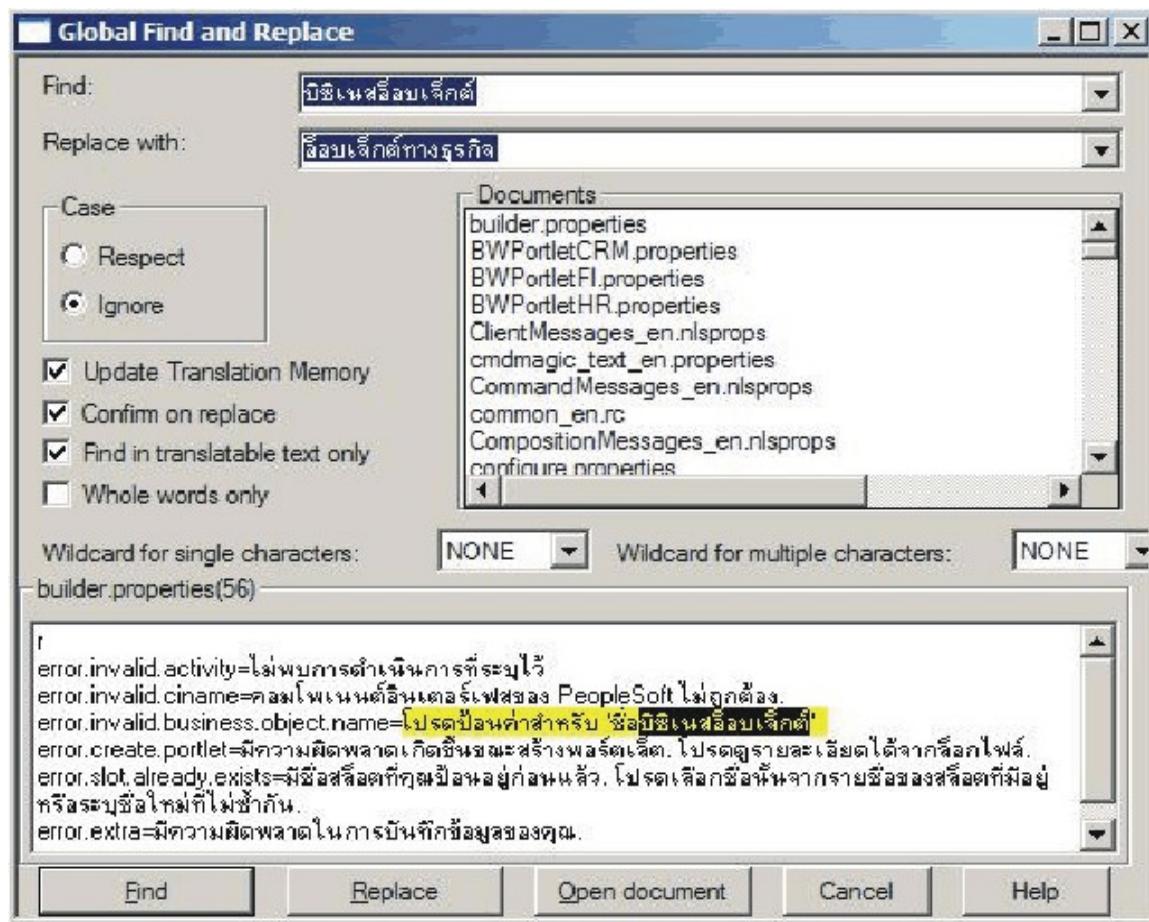


Figure 11. Using the **Global Find and Replace** window with Thai scripts

Using Thai dictionaries and abbreviation lists

You can use Thai dictionaries the usual way, as described in [Working with dictionaries](#). The headwords can be in Thai, as well as the translations. The following dialog box shows an example of how an entry in an English-Thai dictionary is edited. Ensure that the Thai keyboard is selected before editing or adding entries.

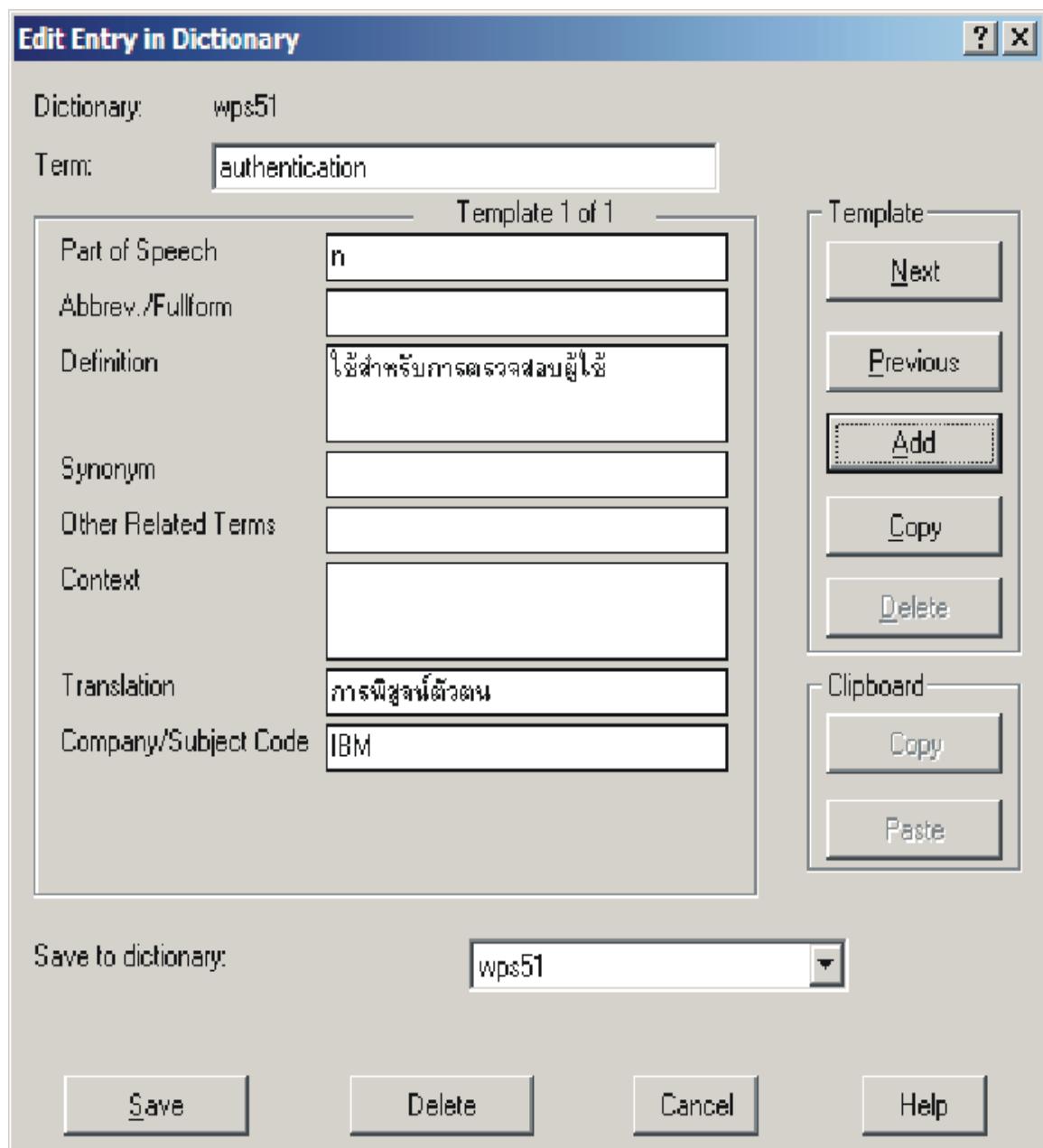


Figure 12. Editing a Thai dictionary entry

You can maintain abbreviation lists, as described in [Maintaining abbreviation lists](#), also for the Thai language. The following dialog box shows an example:

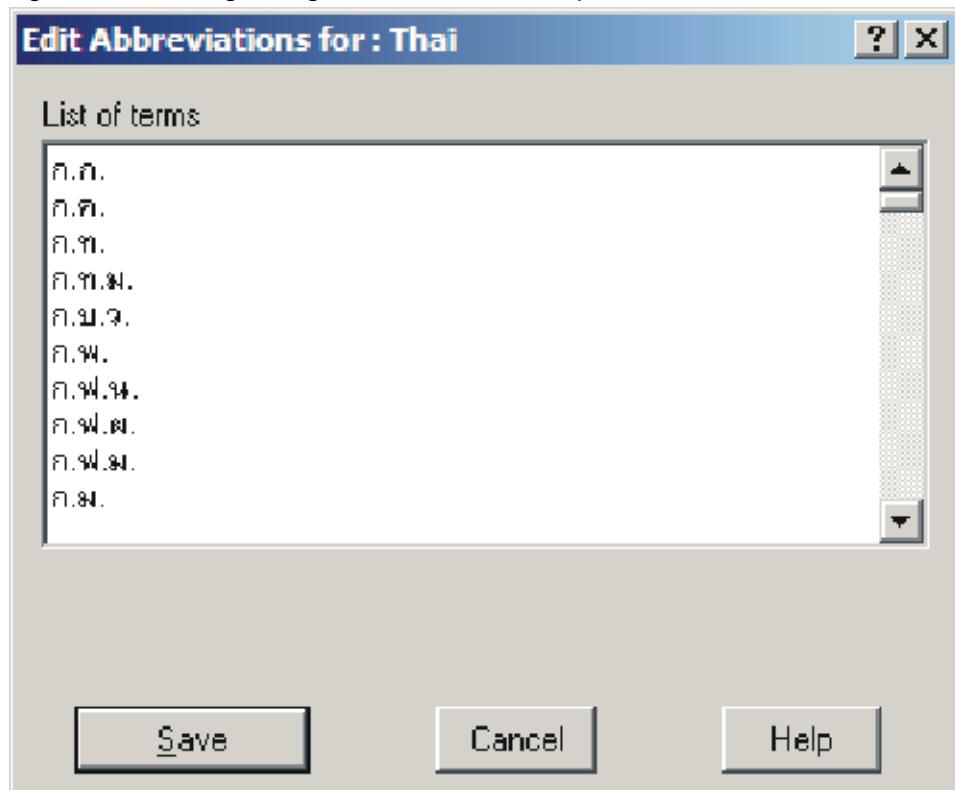


Figure 13. Editing a Thai abbreviation list

Translating HTML documents

The EQFHTML4 markup table is enabled for the Thai language. The following example shows a preview of an HTML document.

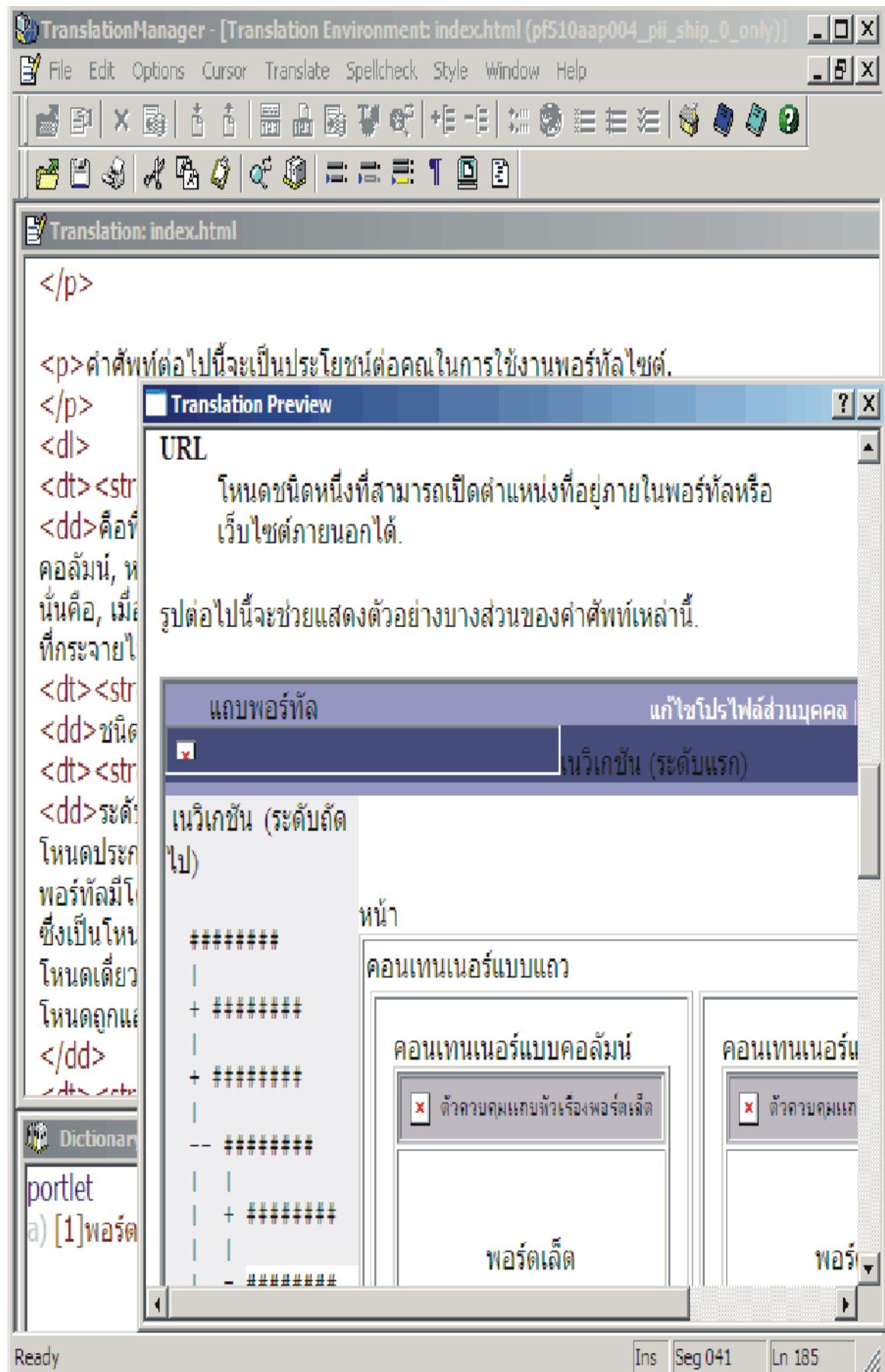


Figure 14. Sample of a Thai translation preview of an HTML document

Limitations of the current Thai support

Note the following limitations of OpenTM2's current Thai support:

- Thai support is only available on Personal Computers that have Thai fonts installed. Code page 874 is supported.
 - The sentence recognition in documents with Thai source language is not always correct.
 - The dictionary lookup dialog box does not display the Thai translation correctly because of the

- fixed pitch size in this dialog box.
- Names of folders, files, translation memories, dictionaries, and terminology lists should not contain Thai characters.
 - The Translation Memory editor cannot display Thai characters. During the creation of an Initial Translation Memory Thai characters cannot be displayed.
 - Further, Thai word counting, spelling check and machine translation are not supported.

Commenting segments in the editor

The segment comments is used to exclude specific memory proposals from the usage in automatic substitution. These memory proposals may contain special translations which make only sense in very rare situations or the target of the memory proposal may not match the source of the proposal because of errors in the document which had to be corrected by the translator.

The comments are entered in the segment properties window in the Translation Manager editor and are saved to the memory when the segment is saved.

The segment comments in the Translation Environment are maintained using the segment properties window. This window is opened using the "Segment properties window" menu item of the "File" pulldown or by assigning a key to the "Segment properties" function and pressing this key.

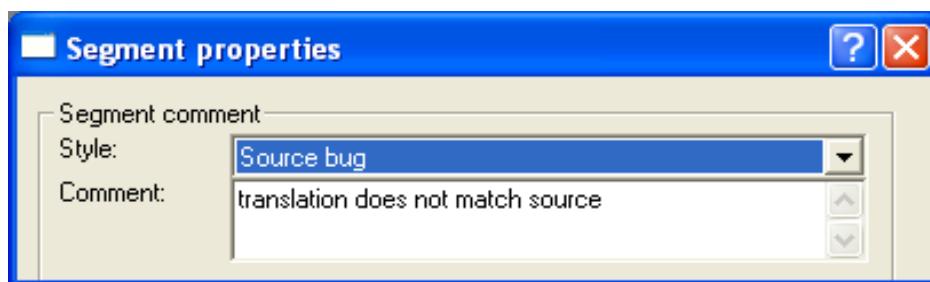


Figure 15. Segment properties window

[\[PIC\]Figure shows Segment properties window](#)

For normal documents the segment properties window shows the comment style and the comment text, for documents in XLIFF folders additional properties are shown. The window is a modeless window: i.e. it can be left open while working with the other windows of the Translation Environment.

The windows always shows the properties of the currently activated segment.

Storing segment comments in the Translation Memory

Whenever a segment with a comment is saved in the Translation Environment (e.g. by pressing Ctrl-Enter) its comment and comment style is saved together with the segment source and segment translation in the Translation Memory.

In the memory proposal window the proposals with comments are prefixed with [Note].

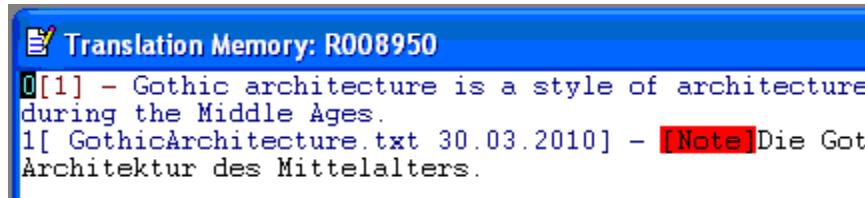


Figure 16. Translation Memory window

[\[PIC\]Figure shows Translation memory window](#)

By double clicking the [Note] indicator the comment of the proposal is displayed in the "Proposal comment" window

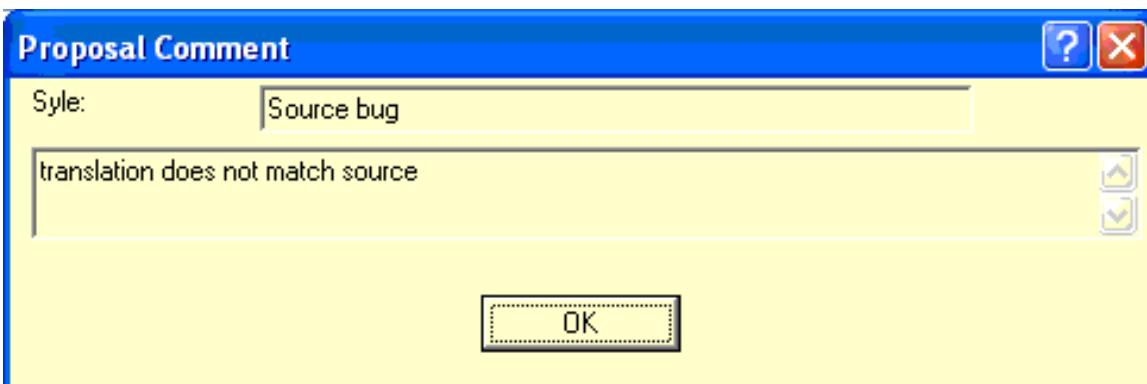


Figure 17. Proposal comment window

[PIC]Figure shows memory proposal comment window

Handling of commented segments in the automatic substitution of the analysis

The new option "Ignore memory proposals with comments" controls the way memory proposals with comments are processed. When this option is selected memory proposals with comments are not used for automatic substitution. The same option is available for the EqfAnalyzeDoc API call and for the EQFBATCH comment line tool.

Working with the OpenTM2 editor

What the OpenTM2 editor is

OpenTM2 has a built-in editor that assists you when translating. It supports all major editing functions and a set of functions that are especially adapted to translations.

You can work with the editor when you have entered the Translation Environment.

RTFEdit editor The descriptions in this chapter apply also to the Rich Text Format editor RTFEdit, which supplements the standard editor in support of Thai language documents. See also [Working with Thai language documents](#), if required.

Basic functions

Usually when you want to do a certain task with OpenTM2's editor, you select a menu that takes you to a window where you fill in the required information. For some basic editing functions OpenTM2 offers you predefined key combinations that you can reassign to suit your individual needs. This is a quick way of performing tasks that you do frequently.

Assigning keys

For some functions you can define your own keys or key combinations by selecting **Keys...** from the **Options** menu. In the "Assign Keys" window select the function you want to define a key for. These are the functions for which no menu commands are available and no keys have been predefined. In the "Assign Keys" window they are indicated by ***none***.

Click **Assign new key**. You are prompted to press the key or key combination you want to assign to a function. Then click **Assign** to save your settings and leave the window.

Clicking **Print** lets you print all functions and their assigned keys.

To remove a key assignment, click **Clear key** in the "Assign Keys" window after you have selected a function.

[Editor functions reference](#) gives you an overview of the tasks to which you can assign keys.

Finding and replacing terms

If you wish to locate a specific term in your translation document, select **Find and Replace...** from the **Edit** menu in the Translation Environment. In the "Find and Replace" window type the word you are looking for in the **Find** field and click **Find**. You can define the following search criteria:

- The spelling of the term to be searched (case sensitivity)
- The direction of the search (forward or backward)
- The text area that can be changed: within the current segment if the document is currently being translated, or throughout the file if the document is currently being revised (post-edited)

A term is found even if it spans over two lines.

If you wish to replace a word by another term, type the new term into the field **Change to** and click **Replace**. The term specified in the **Find** field is changed to the new term in all places where it has

been found in the text. If you wish to change the term only in certain contexts, click **Confirm on changes**. When the term is found press the **Replace** button a second time to replace the found term.

If you wish to locate a certain piece of text but do not know exactly the document where it originates from, you can search through a series of documents related to a folder. See [Searching and replacing text in documents](#) for details.

The "Find and Replace" window remains open until you explicitly close it and is associated with the window that is currently active.

If both the "Translation" window and the "Original" window (source document) are open during a search, the two windows are synchronized. This means that when the searched term is found in the "Translation" window, the "Original" window is scrolled to the segment that contains the source of the searched term, and vice versa.

If you close the "Find and Replace" window and open it again at a later point of time during the same OpenTM2 session or during a new session, the **Find** and **Change to** lists contain the last five terms that you searched for and changed.

Searching for terms in Translation Memory databases

The function **Concordance search** allows to search through one or more Translation Memory databases for terms. This function is useful to search for other translations of the given term or to verify the context of translations.

The concordance search is activated from the **Edit** pulldown menu of the Translation Environment.

The concordance search window is a modeless window which can be opened parallel to the other windows of the Translation Environment; i.e. you can work in the translation window without having to close the concordance search window.

The window can be positioned everywhere on the desktop and its size is customizable.

Prerequisites

The "Translation Environment" is active.

Calling sequence

Select: **Concordance Search...** from the **Edit** menu

The "Concordance Search" window is displayed (see [Figure 1](#))

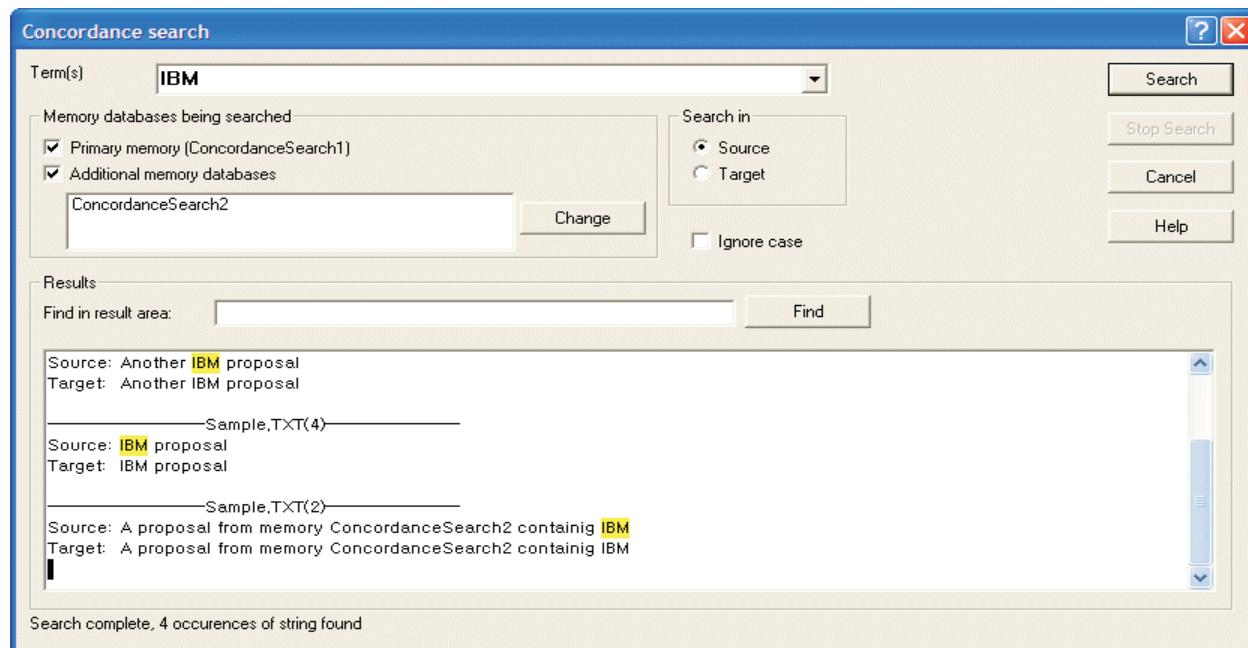


Figure 18. Concordance Search window

[PIC]Figure shows Concordance Search window

Options and parameters

Term(s)

Enter here the term being searched for. This field is preset with the contents of the clipboard. The term is exactly searched like it has been specified in this field.

Source

When selected the search is done in the source of the memory proposals.

Target

When selected the search is done in the target of the memory proposals.

Primary memory

Check this option to search the primary memory. The primary memory is the memory active for the current document. The name of the primary memory is displayed in parenthesis following the text "Primary memory".

Additional Memories

Check this option to select other memory databases. When this option has been active additional memory databases can be selected by pressing the "Change" button.

Change

Press this button to activate the memory selection dialog and select one or more memory databases for the search.

Ignore case

When selected the search is performed case insensitive otherwise is it performed case sensitive.

Results

The search results are shown in this area. Each found proposal starts with a delimiter line containing the document and segment number of the memory proposal followed by the source of the proposal and the target of the proposal. The matching portions of the text are colored in yellow.

Find in result area

Enter here the text which is to be searched in the result area.

Find

Press this button to start the search for the entered text. Each time the button is pressed the next occurrence of the text (if any) is selected. At the end of the result area the search restarts at the beginning of the result area.

Search

Press this button to start the search. The button is only active when some text has been entered in the Term(s) field and when at least one search memory has been selected. The button becomes inactive while the search is performed.

Stop search

This button is only active while a search is running. Press the button to stop the current search and to enter the search criteria for a new search.

Cancel

Press this button at any time to close the concordance search window.

Moving around the text

When you have opened a document you can start using the editor. The text appears in the "Translation" window, and the cursor is placed in the active text unit to be translated (shown with a yellow background). To view another part of the text in the "Translation" window, you must move the cursor. The following table explains how to move the cursor within a segment and within the complete translation and how to scroll text in windows.

Move to...	Action	Comment
Active segment cursormoving	Translate menu, "Go to active segment"	Positions the cursor in front of the first character of the active segment.
Beginning of line	Home	Positions the cursor at the beginning of the current line.
Beginning of segment	Alt+Home	Positions the cursor at the beginning of the current segment.
Bookmark	Translate menu, "Go to bookmark"	Positions the cursor on the bookmark previously set. To go to subsequent one, repeat action.
Bottom of document	Ctrl+End	Moves cursor to bottom of document.
End of line	End	Positions the cursor on the last character of the current line.

Move to...	Action	Comment
End of segment	Alt+End	Positions the cursor on the last character of the active segment.
Newly translated segment	To be assigned	Saves you from scrolling through whole text. Takes you directly to the segment.
Next changed segment	To be assigned	Saves you from scrolling through whole text. Takes you directly to the segment.
Next line	Enter	Positions the cursor at the beginning of the next line.
Next unchanged segment	To be assigned	Saves you from scrolling through whole text. Takes you directly to the segment.
Next untranslated segment	Translate menu, "Next untranslated segment"	Confirms a translation and positions the cursor in the next untranslated segment.
Next word	Ctrl+Right	Positions the cursor on the next word.
Previous word	Ctrl+Left	Positions the cursor on the first character of the previous word.
Tab backward	Backtab	Moves cursor to previous tab stop (tab = 8 characters).
Tab forward	Tab	Moves cursor to next tab stop (tab = 8 characters).
Top of document	Ctrl+Home	Moves cursor to top of document.

Scrolling

Scroll...	Action	Comment
Dictionary proposal down	Alt+down	Moves text 1 line down.
Dictionary proposal up	Alt+up	Moves text 1 line up.
Translation down	Shift+F4	Moves text 1 line down.
Translation left	Shift+F5	Moves text 1 character to the left.
Translation right	Shift+F2	Moves text 1 character to the right.
Translation up	Shift+F3	Moves text 1 line up.
Translation Memory proposal down	Ctrl+down	Moves text 1 line down.
Translation Memory proposal up	Ctrl+up	Moves text 1 line up.
Translation down	Page Down	Moves text 1 page down.
Translation up	Page Up	Moves text 1 page up.

Querying and setting the cursor position

Use the options **Query line** or **Go to line...** from the **Cursor** menu in the Translation Environment to query the line where the cursor is located or to move the cursor to a certain line within your document.

Moving around marked text

To copy, move, or delete a section of text within a segment, the OpenTM2 editor provides key combinations. Selecting a section of the text is called **marking**. You can mark text across several segments or lines. Marked text is displayed in reverse image.

To work with a text section:

1. Mark the text to be copied, moved, or deleted by using the respective key combination. The OpenTM2 editor highlights the area to show you what you have marked.
2. Move the cursor to the position where you want to copy or move the highlighted text.
3. Use the respective key combination depending on what you want to do with the marked text (copy, move, or delete).

To manipulate a block of text, use one of the following key combinations:

Task	Action	Comment
Copy block	To be assigned	Copies a marked text section to the current cursor position.
Copy block to clipboard	Ctrl+Insert	Copies a marked text section to the Windows ^(R) clipboard.
Copy proposal block to clipboard	Ctrl+letter of proposal	copying Translation Memory proposals Translation Memory copying proposals Copies a marked Translation Memory proposal, or part of it, to the active segment in the "Translation" window. Marking is only possible within one proposal. If you specified line wrapping for this translation, the copied text is also wrapped.
Cut block to clipboard	Shift+Delete	Copies a marked text section to the Windows ^(R) clipboard and then removes the block.
Delete block	Key or command to be assigned	Removes a marked text section.
Find block	Cursor menu, "Find block"	Moves cursor to marked text section.
Mark block	Shift+Cursor, or mouse	Marks a text section.
Mark word	Key or command to be assigned	Marks the word where the cursor is positioned.
Move block	Key or command to be assigned	Moves a marked text section to another position.
Paste block from clipboard	Shift+Insert	Inserts marked text section from clipboard into text.
Unmark block	Edit menu, "Unmark block"	Removes marking from text section.

Note:

- Only the active segment or, if in postediting mode, translated segments can be cut to the Windows^(R) clipboard.
- Marked areas can only be moved within the active segment or, if in postediting mode, the translated segments.
- When copying text, remember that a segment can have a maximum size of 2 KB.

Working with bookmarks

OpenTM2 lets you set bookmarks in a text to temporarily indicate a place where you want to return to later.

Task	Key or Menu	Comment
Set bookmark	Translate menu, "Set bookmark"	Sets a bookmark at the current cursor position. Makes it easier to return to this point in text later to complete the translation, for example.
Go to bookmark	Translate menu, "Go to bookmark"	Moves cursor to the position where you set a bookmark.
Clear bookmark	Translate menu, "Clear bookmark"	Removes a bookmark previously set.

Working with lines and words

The OpenTM2 editor provides default key combinations for manipulating lines and words within a document.

Task	Key or Menu	Comment
Delete character <i>editor functionsdeleting characters deletingcharacters</i>	Delete or Backspace key	Deletes the character where the cursor is currently positioned, or the previous character.
Delete line <i>editor functionsdeleting lines deletinglines</i>	To be assigned	Deletes the line where the cursor is currently positioned.
Delete until tag <i>editor functionsdeleting text until next markup tag deletingtext until next markup tag</i>	To be assigned	Deletes all characters from the current cursor position up to the next tag.
Delete word <i>editor functionsdeleting words deletingwords</i>	To be assigned	Removes the term where the cursor is located.
Find and replace <i>editor functionsfind and replace terms finding and replacing terms</i>	Edit menu, "Find and replace"	Locates the required term and replaces it as specified.
Insert line <i>linesinserting inserting a line Insert line editor functionsinserting a line</i>	To be assigned	Inserts a line after the line where the cursor is currently positioned.
Join lines <i>linesjoining joining lines Join lines editor functionsjoining lines</i>	To be assigned	Joins the current line with the following one.
Split line <i>linesplitting splitting a line Split line editor functionssplitting lines</i>	To be assigned	Splits the line at the current cursor position.
Truncate line <i>linestruncating truncating lines Truncate line editor functionstruncating lines</i>	Ctrl+Delete	Deletes all characters from the current cursor position until the end of the line.
Undo <i>undoing an action Undo editor functionsundoing an action</i>	Alt+Backspace	Restores the original state as it was before an action was performed. You can undo up to five preceding operations.
Wrap line <i>lineswrapping wrapping lines Wrap line editor functionswrapping lines</i>	Edit menu, "Line wrap (toggle)"	Set wrapping at margin column on or off.

Translation-specific functions

The OpenTM2 editor particularly supports you when you translate in the Translation Environment.

Task	Key or Menu	Comment
Add an abbreviation <i>abbreviationsadding new ones</i>	Translate menu, "Add an abbreviation"	Adds the term where the cursor is positioned to your abbreviation list.
Automatic substitution <i>automatic substitution</i>	Translate menu, "Automatic substitution"	Replaces segments for which a translation has been found in the Translation Memory by their previous translation. Automatic substitution stops when no exact match or more than one exact match has been found.
Change display style of markup tags	Style menu, "Protect", "Unprotect", "Hide", "Shrink", "Compact"	Displays the markup information in a document according to the selected style. To avoid overtyping by mistake, all markup tags are protected by default.
Copy dictionary proposal <i>dictionarycopying proposals copyingdictionary proposals</i>	Ctrl+letter of proposal	Inserts the translation proposal at the current cursor position. The first 26 dictionary entries are preceded by a to z, the

Task	Key or Menu	Comment
		entries 27 to 52 are preceded by A to Z. Entries beyond 52 are preceded by an asterisk (*) and must be copied from the clipboard using Copy and Paste.
Copy Translation Memory proposal <i>copying Translation Memory proposals</i> <i>Translation Memory copying proposals</i>	Ctrl+number of proposal	Replaces the active segment with the translation proposal. See for information on how to copy a marked part of a translation proposal.
Delete Translation Memory proposal <i>deleting Translation Memory proposals</i> <i>Translation Memory deleting proposals</i>	"Delete" key while cursor is on proposal	Lets you remove translation proposals from the Translation Memory, for example, translations that have been added by mistake.
Display exact fuzzy proposals	To be assigned	Displays fuzzy translation proposals in addition to an exact one in the "Translation Memory" window. Is useful for viewing more than one translation possibility for a segment.
Edit abbreviations <i>abbreviations updating</i>	Translate menu, "Edit abbreviations..."	Enables the extension and update of abbreviations defined for the selected language-support file.
Edit addendum terms <i>spellchecker addendum, updating</i>	Spellcheck menu, "Edit addendum terms..."	Lets you modify or remove terms which have been added to the spellchecker addendum (for example, terms that have been added by mistake).
Edit an entry in a dictionary	Translate menu, "Edit a term...", or positioning cursor on term in "Dictionary" window and double-clicking mouse button 2	Lets you edit a found entry or add a new one while translating. To edit a multiword term, all components of the term must be marked first.
Look up a sentence in the Translation Memory <i>looking up sentences in a Translation Memory</i> <i>Translation Memory looking up sentences</i>	Options menu, "Sentence lookup..."	Looks up a sentence in the Translation Memory while you translate.
Look up a term in a dictionary <i>looking up terms in a dictionary</i> <i>editing dictionary entries</i> <i>dictionary looking up terms</i> <i>dictionary editing entries</i>	Translate menu, "Look up a term...", or positioning cursor on term in "Dictionary" window and double-clicking mouse button 1	Looks up entry of term where the cursor is positioned. To look up a multiword term, all components of the term must be marked first.
Postedit <i>postediting</i> <i>translations revising</i>	Translate menu, "Postediting"	Lets you revise what you translated. Whatever you change in the translation is automatically changed in the Translation Memory.
Spellcheck <i>translations spellchecking</i>	Spellcheck menu, "File" or "Segment"	Checks the spelling of a translated document or an individual segment.
<i>toggling first character of dictionary</i> Toggle first character of dictionary proposal <i>editor functions toggling first character of dictionary proposal</i> <i>dictionary toggling first character of proposal</i>	Options menu, "Keys..."	Changes the first character of the dictionary proposal from uppercase to lowercase, and vice versa. If the dictionary proposal consists of several words, it changes the first character of the word where the cursor is placed. This function does not apply to DBCS characters.

Task	Key or Menu	Comment
View another document <i>translationsreferring to other documents while translating</i> <i>documentopening more documents while translating</i>	File menu, "Open..."	Opens another document for checking purposes while translating. Parts of this document can be marked for copying into the document you currently translate.
View original documentdisplaying original while translating <i>original document, viewing</i>	File menu, "Original window"	Displays the source document while translating.
View source of a translation proposal <i>Translation Memory matchesdisplaying corresponding sources</i> <i>matchesdisplaying corresponding sources</i>	Filemenu, "Source of Proposal", or double-clicking mouse button 1 while being in proposal window	Displays the source of a translation proposal.
View table of contents <i>Translation Environmentdisplaying tables of contents</i> <i>documentdisplaying tables of contents</i> <i>viewingtables of contents</i> <i>displayingtables of contents</i> <i>table of contentsdisplaying</i>	Cursor menu, "Special go to..."	Displays a table of contents of the document being translated. Double-clicking an item directly takes you to this item in the document. To enable the display, you must include a specific tag in your markup table (see classid).
View translation <i>Translation Environmentdisplaying translations</i> <i>documentdisplaying translations</i> <i>viewingtranslations</i> <i>displayingtranslations</i>	Translate menu, "Show translation"	Displays the translation document in HTML, RTF, or Microsoft ^(R) Word format. Prerequisite is the availability of the WebBrowser control.
View translator's notes <i>Translation Environmentdisplaying translator's notes</i> <i>documentdisplaying notes</i> <i>viewingtranslator's notes</i> <i>displayingtranslator's notes</i>	To be assigned	Displays the notes for the translator that are contained in the source document.

Manipulating segments

OpenTM2 divides the translation into individual text units (segments). The editor provides functions to conveniently work with these segments.

Task	Key or Menu	Comment
Compress segment <i>segmentscompressing</i>	To be assigned	Turns segments presented in a style other than compact into compact style again.
Delete segment <i>segmentsdeleting</i>	To be assigned	Deletes the active segment.
Expand segment <i>segmentsexpanding</i>	To be assigned	Turns a segment into protected style if it was compact before.
Join segments <i>segmentsjoining</i>	Translate menu, "Join segments"	Combines the active segment with the following one, and changes the way a document file is segmented.
Mark segment <i>segmentsmarking</i>	Key or command to be assigned	Marks the active segment.
Reflow segment <i>segmentsreflowing</i>	Edit menu, "Reflow segment"	The current segment is reflowed (dependent on definition of right margin). This function is not active if right margin is set to AUTO.
Spellcheck segments <i>segmentsspellchecking</i>	Spellcheck menu, "Segment"	Checks the spelling of a segment.
Split joined segments <i>segmentssplitting</i>	Translate menu, "Split joined segments"	Splits up previously combined segments at the current cursor position.

Task	Key or Menu	Comment
Translate segment segmentstranslating	Translate menu, "Translate segment" or Ctrl+Enter	Confirms a translation and activates next segment. Alternative way to activate any segment in document where cursor has been positioned. Can be used to leave postediting mode.
Truncate segment segmentstruncating	Alt+Delete	Deletes all text from the current cursor position to the end of the segment.
Untranslate segment segmentsuntranslating	Translate menu, "Untranslate segment"	Deletes the translated segment from the Translation Memory. The original segment replaces the translated one.

Manipulating documents

The following functions let you save your translation, print it, and leave the Translation Environment. There is also an option to spellcheck the complete translation.

Task	Action	Comment
End-Save translationssaving	File menu, "End-Save" or F4	Saves the translation and leaves the Translation Environment.
Next	File menu, "Next"	Toggles between original, translation, and any other open document.
Open	File menu, "Open..."	Opens another document.
Previous	To be assigned	Toggles between original, translation, and any other open document in the ring.
Print translationsprinting printingdocument displayed in Translation window	File menu, "Print"	Prints the document displayed in the "Translation" window.
Quit	File menu, "Quit" or F3	Closes the Translation Environment and returns you to the Translation Workbench.
Save	File menu, "Save" or F2	Saves the translation document.
Spellcheck file	Spellcheck menu, "File"	Checks the spelling of the document displayed in the "Translation" window.

Spellchecking

To proofread an entire translation document or just a segment, select **File...** or **Segment...** from the **Spellcheck** menu while you are in postediting mode. Click **Skip** when misspelled words are indicated (in the **Misspelled word** field) that you consider correct or unimportant.

If a word turns out to be misspelled, either double-click the right word in the **Similar words** list box or type the correct word in the **Change to** field and click **Change**. The misspelled word is then replaced with the correct word.

To permanently add the correct word to the spellchecker, click **Addendum**. If you want to add this word only temporarily, click **Temp add**.

This addendum can be edited at a later time. For example, to remove, change, or add more terms.

If no more misspelled words are found, you receive a message.

To practise handling the proofreading aid, refer to the *Translator's Workbook*, Lesson 11: Postediting translations.

For more information on spellchecking, refer to [Spellchecking a document](#).

Viewing other documents while translating

If you wish to view another document while translating, select **Open...** from the **File** menu.

For a document already imported in OpenTM2, you can specify whether you want to browse the translation or the original. Then select the folder that contains the document from the **Folders** list. In the **Documents** list select the requested document and click **Open**.

For an external text file, first select the drive where this file is stored and the respective directory from the **Directories** list. In the **Files** list select the requested file and click **Open** to view it.

To leave this window, click **Cancel**.

Configuring the editor

The following options help you tailor the Translation Environment and make translating easier for you.

Customizing colors and fonts

You can change the font and font size of the text in the various windows of the Translation Environment and the color of the various text types and markup tags to suit your needs.

If you are working with Thai language documents, see [Working with Thai language documents](#) for customizing colors and fonts.

To change the font and font size, choose **Fonts** on the **Options** menu. This takes you to the "Set Fonts" window on which you can select the window for which you want to change the font and font size. In the **Sample text** box you see an example of how the changed font looks. When you are satisfied with your changes, click **Set** to save them. To get the initial settings again, click **Defaults**.

To change the color, choose **Colors** on the **Options** menu. This takes you to the "Set colors" window on which you can select the text type for which you want to change the foreground color, background color, or both. In the **Sample text** box you see an example of how the changed color looks. When you are satisfied with your changes, click **Set** to save them. To get the initial settings again, click **Defaults**.

Refer to the lesson on customizing the Translation Environment in the *Translator's Workbook* to learn how to change fonts and colors.

Displaying a ruler

Sometimes it is necessary to position text at a specific place in a translation. For this purpose, OpenTM2 enables you to display a ruler in the "Translation" window or the window that contains the original document or another document. The ruler has the same size as the currently selected font.

To display a ruler in the "Translation" window, select **Tailor** from the **Options** menu or the context-sensitive popup menu and choose **Ruler**. To display a ruler in the window containing the original document or another document, select **Tailor** from the context-sensitive popup menu and choose **Ruler**. You get the context-sensitive popup menu by pressing the right mouse button.

Displaying the cursor position

OpenTM2 enables you to display the current position and status of the cursor while you are translating:

Select **Tailor** from the **Options** menu or the context-sensitive popup menu and choose **Statusbar**. In the bottom right corner of your "Translation" window, OpenTM2 displays the following information:

- Whether you are working in insert or replace mode (**Ins** or **Ins**)
- The number of the segment (**Seg**) you are currently translating
- The number of the line (**Ln**) in which your cursor is positioned
- The number of the column (**Col**) at which your cursor is positioned

Setting up the profile

You can customize some translation-specific functions in the editor according to your needs. Do this as follows:

- Open the **Options** menu
- Select **Profile settings...**

This takes you to the "Customize Translation Functions" window. The options you specify in this window become immediately active when you continue working with [OpenTM2.windowsCustomize](#)

Translation Functions options

You can customize the following options:

Customize Translation Functions Editor options

cursorswitching to insert mode

- On the “Editor” page:
 - **Set Insert mode for active segment**
If you select this check box, the cursor state is switched to insert mode each time you start translating a new segment, even if it was in replace mode before.

editor functions Backspace key

Backspace key

- **Insert proposal if inset mode is active**
If you select this check box, a proposal that you copied from the “Translation Memory” window is inserted at the current cursor position when the cursor is in insert mode. When the cursor is in replace mode, and this check box is cleared, the proposal replaces the active segment.
- **Visual display**
Determines the display of embedded text in bidirectional language documents. Select this check box if you want embedded numerals and text in an Arabic or Hebrew document to be shown from right to the left. Clear this check box if you want these parts displayed from left to the right. See also [Working with bidirectional language documents](#), if required.
Have this check box cleared also for all non-bidirectional language documents.

shift-in/shift-out characters, automatic insertion of

DBCS, automatic insertion of shift-in/shift-out characters

double-byte characters, automatic insertion of shift-in/shift-out characters

- **Automatic insert of SO/SI characters**
If you select this check box, shift-in (SI) and shift-out (SO) characters are inserted for double-byte characters (Asian languages).
SO characters and SI characters are treated as protected characters to prevent them from being overtyped. They are still displayed in compact style mode.

Customize Translation Functions Display options

- On the “Display” page:
- **Visible white spaces**
If you select this check box, you can select a character that is displayed instead of a blank or a line feed.
 - **Visible blank**
Select a character from this drop-down list that you want to be displayed instead of a blank.
 - **Visible line feed**
Select a character from this drop-down list that you want to be displayed instead of a line feed.

Notes:

- Some of the selectable characters might not display properly in the active system language.
- Do not select the same character for both options, otherwise you cannot distinguish for what the substitution character stands.
- You can choose a blank as a substitution for one or the other option. However, choosing a blank for *all* options causes the Translation Environment to appear as if the check box is not selected. Avoid this combination.
- If you use the RTFEdit editor, the selections for **Visible blank** and **Visible line feed** are disabled because standard Windows characters are used for both selections.

RTFEdit editor visible blank characters visible line feed characters

- **Segment boundary sign in post edit**
If you select this check box, you can select a character that is displayed at the end of each segment when you postedit your translation.
- **Segment boundary sign**
Select a character from this drop-down list that you want to be displayed at the end of each segment.

Note: See also the previous notes.

- **Shrink-style abbreviation**
Displays the symbol used as a placeholder for the information that cannot be translated. This symbol is displayed in the “Translation” window if you select the shrink style from the **Style** menu. You can overtype this symbol with a symbol of your choice. This symbol can consist of up to three characters, for example </>.
- **Compact-style abbreviation**
Displays the symbol used to mark information in a segment that cannot be translated. This symbol is displayed in the “Translation” window if you select the compact style from the **Style** menu. You can overtype it with a symbol of your choice. It can consist of up to three characters, for example </>.
- **Note**
Displays the name of the markup tag in the source document indicating that this document contains one or more notes. You can overtype this name with a name of your choice.

- **Line number or active segment**
Displays the number of the line containing the active segment. You can change it by specifying a number from 1 to 99.
- **Right margin**
Determines the line length of a translated segment. You can select **Auto** to let the line length automatically adapt to the window size. You can also select one of the offered numbers or specify a number of your choice. This number determines how many characters are to fit a line.
This option only applies to segments still to be translated, not to segments that are already translated.
- **Customize Translation Functions**[Dictionary options](#) **TEXT**
- On the "Dictionary" page:
 - **Information for dictionary terms**
If you mark this option, the "Dictionary" window contains additional information about each dictionary entry. You control which dictionary entry fields are displayed by selecting or deselecting, for each field, the **Aut.Lookup** check box in the "Dictionary Properties" window. If too many dictionary entry fields are to be displayed, they are abbreviated in the "Dictionary" window.
 - **Dictionary terms of all dictionaries**
If you mark this option, the "Dictionary" window displays all translations that are found in the folder dictionaries for the terms in the current segment. If you do not mark this option, only the first translation found for the terms is displayed.
 - **Dictionary indicator**
If you mark this option, the dictionaries listed in the title bar of the "Dictionary" window are numbered. In this case, each translation of a term is preceded by the number of the dictionary from which it comes. The maximum number of dictionaries allowed is 10, where the tenth dictionary is preceded by the letter A.
The default color of the dictionary indicator is gray on a white background. You can change these colors in the "Set Colors" window (see [Customizing colors and fonts](#)).
 - **Show single words of compound terms in auto-lookup**
If the option is active the single words of a compound term are looked up together with the compound term. If the option is inactive the single words of a compound term are not looked up in the dictionary auto-lookup.
An example:
Assume a dictionary containing the terms "data", "data processing", "data processing unit", and "unit". If a segment contains the term "data processing unit" the dictionary window will show the translations for "data", "data processing", "data processing unit", and "unit" if the option is active and "data processing unit" if the option is inactive.
- **Customize Translation Functions**[Translation Memory options](#)
- On the "Transl.Mem" page:
 - "Translation Memory window"
Lets you choose how the markup information is to be displayed in this window. You can choose between Protect, Hide, Compact, and Compact+1. Protect is the default. For more information on these styles, refer to [How markup tags are displayed](#).
 - "Display 'Source of Proposal' window"
If you select this check box, the "Source of Proposal(s)" window is always displayed when you work in the Translation Environment. If you clear this check box, the window only appears when you double-click in the "Translation Memory" window, and it disappears again when you continue translating the next segment.
 - **Number of proposals**
If you select this check box, the number of found translation proposals is displayed in front of the original segment in the "Translation Memory" window and the "Source of Proposal(s)" window. For example, [2] in either of these windows means that two translation proposals are available.
 - **Document of proposal**
If you select this check box, the file name of the document where a translation proposal originates from is displayed in front of the proposal in the "Translation Memory" window and the "Source of Proposal(s)" window. For example, [DEVICE.HTML] means that the displayed proposal originates from document device.html.
 - **Date of proposal**
If you select this check box, the date when a translation proposal has been stored in a Translation Memory is displayed in front of the proposal in the "Translation Memory" window and the "Source of Proposal(s)" window. For example, [DEVICE.HTML 3/21/00] means that the displayed proposal has been saved on the date shown.

Translation Memory matches[displaying MT matches](#)
[matches displaying MT matches](#)

- **Always display MT match**
If you select this check box, MT matches are always displayed in the "Translation Memory" window. If you clear this check box, MT proposals are only displayed if there are no fuzzy proposals, or fuzzy proposals of poor quality.

Translation Memory matches[having terms replaced automatically](#)
[matches having terms replaced automatically](#)

- **Automatic replacement in fuzzy matches**
If you select this check box, all information that is language independent (and therefore

need not be translated), such as dates and proper nouns, is automatically replaced in a fuzzy match to reflect the information in the active original segment. This results in more precise proposals. It also saves you from revising the proposal after it has been copied from the "Translation Memory" window.

This option is particularly useful if you translate documents that contain numerous dates, figures, and abbreviations (for example, stock market reports).

To show the automatic replacement, the proposal is preceded by an [r] when it is displayed in the "Translation Memory" window.

The following example shows an automatic replacement within a fuzzy match:

Current original segment:

(0) - Windows 98 Developer's Toolkit

Similar segment in a previous original document:

(0) - Windows 95 Developer's Toolkit

This segment has previously been translated into German:

Windows 95 Entwicklungswerkzeug

If you have not marked this option, you get this translation offered as a proposal in the "Translation Memory" window: 1[f] - Windows 95 Entwicklungswerkzeug

If you have marked this option, the match is presented as follows:

1[r] - Windows 98 Entwicklungswerkzeug

The version number (95) has been updated to match the current original segment.

Note:

A translation proposal can be fuzzy in some parts and also contain information that has automatically been replaced (for example, a date). This proposal would be preceded by [rf].

editor functions abbreviating Translation Memory proposals

Translation Memory abbreviating proposals

abbreviations Translation Memory proposals

- **Abbreviate Translation Memory proposal**

If you select this check box, the original segment in the "Translation Memory" window (preceded by [0]) is truncated if it is longer than one line. This is shown by an ellipsis (...).

If this check box is cleared, the "Translation Memory" window displays the complete original segment even if it exceeds one line.

editor functions display all exact proposals

proposals display all

translation proposals display all

Translation Memory display all proposals

- **Display all exact proposals**

If you select this check box, the "Translation Memory" window permanently shows up to nine available translation proposals for a current segment. If multiple proposals are available, they are displayed in chronological order. Proposals from the current document precede those from other documents.

If this check box is cleared, only one (the most recent) translation proposal is displayed for a segment. If more proposals are available, this proposal is preceded by [1++]. You can actuate the display of (up to nine) further proposals by double-clicking ++ in the "Translation Memory" window, or by pressing an assigned shortcut key. However, you need to repeat this for every segment.

translation proposal quality of proposal

proposals quality of fuzzy matches

Translation Memory quality of fuzzy matches

- **Quality of Proposal (in %)**

If you select this check box, the "Translation Memory" window permanently shows a percentage number with every fuzzy translation proposal. Percentage numbers rank fuzzy proposals; the topmost proposal has the highest rank.

If this check box is cleared, no percentage number is displayed. Nevertheless, the topmost proposal is likely to fit best.

Customize Translation Functions Messages options

markup tags checking for changes in segments

- On the "Messages" page:

- **Check for changes of inline tags**

If you mark this option, OpenTM2 checks if you changed, deleted, or added any markup tags within segments. For example, it checks if you changed a tag that indicates a change of the font type used. If a tag has been changed, you get a warning message. This ensures that the layout of the translation document still looks like that of the original document, and no markup information is changed. If you do not mark this option, the warning message is not displayed.

editor functions suppressing messages

messages suppressing

suppressing messages

- **Suppress message 'Fuzzy proposal unchanged'**

If you do not mark this option, you get a warning message when you save a segment that you copied from a fuzzy proposal and did not change it (this is the default). This helps you remember to revise segments that you translated using a fuzzy translation proposal. If you

mark this option, the message is not displayed.

- **Suppress message 'Source unchanged'**

If you do not mark this option, you get a warning message when you want to save an untranslated segment as a translation. If you mark this option, the message is omitted, and the original segment is saved as a translation. This might be useful when the original segment contains information that you do not want to translate (for example, citations, time and date information).

- On the "Aut. Substitution" page:

[Customize Translation Functions](#)
[Aut. Substitution options](#)

- **automatic substitution interrupting**

- **Interrupt automatic substitution**

Select this check box if you want the automatic substitution to stop if a segment cannot be substituted automatically. This occurs when OpenTM2 cannot find an exact match in the Translation Memory or if it finds more than one exact match. If you clear this check box, OpenTM2 skips all segments for which no exact match is found or for which more than one exact match is found and continues automatic substitution.

- **Use exact context match when more than one found**

Select this check box if you expect segments with more than one exact match in the associated **Translation Memory** and if you want the system to choose the one from the same document. If you clear this check box, OpenTM2 skips the automatic substitution for this segment whenever more than one exact translation is found for a source segment.

- **Use latest exact match when more than one found**

Select this check box if you want an automatic substitution run not to stop when more than one exact match is encountered. The most recent match in the Translation Memory is taken for the substitution.

- **Adjust leading white space to white space in source segment**

Select this check box if you want the text to have the same leading white space (blank, tab stop, and line feed) before and after the automatic substitution, or when copying a proposal from the Translation Memory in the active segment.

No adjustment is the default.

- **Adjust trailing white space to white space in source segment**

Select this check box if you want the text to have the same trailing white space (blank, tab stop, and line feed) before and after the automatic substitution, or when copying a proposal from the Translation Memory in the active segment.

No adjustment is the default.

Sharing a profile

You may want to provide all translators or those working on the same project with the same profile settings and shortcuts. You can do this in the following way:

1. Prepare all settings on one workstation.
2. Stop **OpenTM2** on all the workstations to receive the shared profile.
3. Copy the file `eqfprow.prp`, which contains all settings, from directory `\eqf\property` to a diskette or a drive shared among all translators.
4. Put the copied file in the `\eqf\property` directory of each translator's workstation.

Using OpenTM2 in the translation business

Translation scenarios

OpenTM2 is designed to help you perform your daily translation work. As a flexible tool that can be adapted to your specific requirements it improves your productivity regardless of the types of documents you translate and the size of your business.

The following scenarios give you an overview of how you can perform your translation tasks using OpenTM2. They show you how to:

- Translate a new document
- Translate an updated document
- Use OpenTM2 when a project involves only one translator
- Use OpenTM2 when a project involves several translators
- Exchange data with other translators and systems

Translating a new document

Translating a new document involves the following tasks:

- Creating a folder
- Importing the new document
- Translating the document
- Postediting the document

- Exporting the document

These tasks are briefly described in the following.

Creating a folder

New documents must be included in a folder before you can translate them. Therefore your first step is to decide whether you want to create a new folder or include the new document in an existing folder. The folder concept allows you to group the entire translation material belonging to a specific project into one folder. If the new document does not belong to an existing project, create a new folder. A folder can contain only one document or several documents.

When you create a new folder, choose a folder name that describes the subject of the project and is easy to remember.

Each folder is associated with a Translation Memory. You can either specify a new Translation Memory or use an existing one. You can use a new Translation Memory for each folder. If you later decide that several folders should share one Translation Memory, you can merge the Translation Memory databases of these folders.

You can also specify whether you want to create a new dictionary with the folder or use one or more existing dictionaries. If you specify existing dictionaries, OpenTM2 searches these dictionaries during translation and displays the translated terms for you to include them in the text.

Importing a new document

When you import a new document into the newly created or an existing folder, the document is analyzed, that is, its text is broken into segments. You later translate the document segment by segment. Each translated segment is stored in the Translation Memory associated with the folder or the specific document. If a segment appears a second time in the document or in another document of the folder, OpenTM2 displays the translated segment to you. You then only have to include the displayed translated segment in your translation. In this way, you achieve consistency throughout your document.

Translating the document

When you open a document for translation, OpenTM2 displays the document with the first segment being highlighted. You translate the segment by overtyping the source text and using the full-featured editor provided by OpenTM2. In addition to the document you also see a "Translation Memory" window, which displays translation proposals, and a "Dictionary" window, which displays dictionary terms (single words or word groups). You can copy the proposals and terms into your document.

While you are translating a document, you can open another document in **OpenTM2** or files outside **OpenTM2**. OpenTM2 enables you to copy text from the opened document into your current translation.

Postediting and viewing the translated document

After you have translated all segments of the document, you can move directly into postediting mode, which allows you to revise and correct your translation. In postediting mode, the "Translation Memory" and "Dictionary" windows are not displayed, and you can type text anywhere in the document.

In addition, you can display the translated document in HTML or RTF format.

Exporting the translated document

After translating and revising the document, you can export it back into its original word-processing format.

Translating updated documents

To translate an updated document, you first import it into a folder referencing the Translation Memory containing the old translation (of the original document). You then have the following options:

- Let OpenTM2 automatically translate the unchanged parts during analysis. You then only have to deal with the new and untranslated text.
- After analysis, open the changed document and run the automatic substitution. OpenTM2 then stops at the places where your interaction is required, for example, where you have to translate the new parts or where you must decide which match to take if the Translation Memory supplies several matches.
- After analysis, open the changed document and work through it segment by segment, copying the translation for the unchanged parts into the changed document and translating the changed parts.

After the translation is complete, you can postedit the translation regardless of which option you chose. The associated Translation Memory is updated automatically.

When you export a changed document, you can ask OpenTM2 to place different revision marks next to the old text, the changes, and the text that you copied from the Translation Memory.

Translating with one translator

When a translation project involves only one translator, this person must also carry out the management and terminology tasks that support the translation process. These tasks are performed under Translation Workbench, whereas the translation itself takes place in the Translation Environment.

Working with the Workbench

The Workbench consists of the following parts: folders, dictionaries, Translation Memory databases, and three kinds of terminology lists. It enables you to perform the various tasks, such as:

- Create a new dictionary or access an existing one for maintenance purposes such as modifying, adding, or deleting entries
- Merge Translation Memory databases
- Process terminology lists
- Import and export documents

From the Workbench you can also open a document to access the Translation Environment.

Working with the Translation Environment

After opening a document, you are in the Translation Environment where you translate segment by segment by using the Translation Memory and dictionary proposals. The original document is still in the background and can be viewed at any time. After postediting the translation, you return to the Workbench to export the document into its original word-processing format.

Translating with several translators

OpenTM2 can also be used by a team of translators. This is useful in particular if a large document has to be translated. You can divide the large document into several parts and include these parts in individual files. In addition, you can assign one person the role of a coordinator who performs all duties associated with the preparation of the translation material. These duties can include maintaining and importing dictionaries, making sure that each team member receives the appropriate dictionary for the document to be translated, and performing all Workbench-related tasks.

When the Workbench is handled by a coordinator, translators perform their work exclusively in the Translation Environment. All resources needed for translation can be prepared beforehand, and the only functions the translators need to select from the Workbench are the Open folder and Open document options.

Due to the split of OpenTM2 into a Workbench and a Translation Environment, work can easily be assigned to a vendor. The coordinator can supply the vendor with a folder containing all the required resources, namely the Translation Memory databases, dictionaries, and documents to be translated. After receiving the translated folder, the coordinator can merge the Translation Memory databases returned by the vendor into a main Translation Memory for future translations. The dictionaries updated by the vendor can also be merged into one main dictionary and then be maintained by the coordinator or project leader.

In addition, the word count utility enables you to count all the words to be translated and so facilitates billing.

Sharing translation resources

To facilitate teamwork OpenTM2 lets you share a dictionary or a Translation Memory with fellow translators working in the same LAN (Local Area Network) system. This way of sharing resources not only helps keeping your own used disk space small but also makes it possible to use a Translation Memory or a dictionary even if another translator is using it at the same time. You may find this especially convenient when you work together with fellow translators on the same project.

For example, if you must translate a sentence that your colleague has translated only recently for the same project, you can instantly use your colleague's translation by copying it from the shared Translation Memory. For detailed information on sharing dictionaries and Translation Memory databases, see [#unique_131](#) and [#unique_132](#).

Exchanging data with other people and systems

OpenTM2 lets you exchange translation material with other OpenTM2 users by means of its export and import facility. For this purpose OpenTM2 uses a specific exchange format (based on SGML tags), except when exporting or importing a folder or document. This feature ensures a consistent interface with the outside world, such as your own terminology databases and full-text editors.

Exchanging documents

When a translation is completed, it must be exported from OpenTM2 and loaded into its original word-processing system for final formatting and printing. The markup information is preserved throughout the document because all tags are protected.

An exported translation can be copied on a diskette or forwarded to another translator or a word-processor operator using e-mail.

Exchanging dictionaries

You can use your own terminology databases or dictionaries in OpenTM2 using the SGML-based exchange format. You must add SGML tags to the file containing your database or dictionary to make it compatible with OpenTM2.

OpenTM2 dictionaries can also be converted into other formats. You first export them as external dictionaries and then replace the SGML tags with the appropriate coding. Outside OpenTM2, the dictionaries can be copied on a diskette or any other media for distribution.

OpenTM2 dictionaries can also be distributed as part of a folder. This is convenient for copyright-protected dictionaries that can only be transferred from one workstation to another using the folder export and import procedures.

Exchanging terminology lists

Terminology lists can be exported as standard text files and processed with any editor or word processor. They can also be distributed in this format on a diskette or any other media.

Terminology lists can be reimported into OpenTM2 for dictionary maintenance.

Exchanging Translation Memory databases

Translation Memory databases can be created from scratch using the Initial Translation Memory utility. If you have a document and its translation in external format, **OpenTM2** can align the translated segments with the original segments. In this way you can create a Translation Memory to be used in OpenTM2.

Translation Memory databases can be exported to an external SGML-based format and copied on diskettes or any other media for distribution and modification. They can be imported back onto another workstation through a standard import procedure. They can also be merged into an existing Translation Memory.

Exchanging folders

Folders cannot be exported to an external format. Exported folders remain in OpenTM2 format with the data being compressed to save storage. When you export a folder, you can choose which of the documents it contains should also be exported. In addition, you can decide whether the associated dictionaries and Translation Memory databases should be exported with the folder. For future reference, you can add a note describing the contents of the exported folder. Folders can be copied on a diskette or any suitable media for distribution.

Exchanging markup tables

OpenTM2 is an open platform that supports many word-processing formats by using specific markup tables. These markup tables enable OpenTM2 to distinguish between formatting information in a document and actual text to be translated.

You may want to distribute a markup table to someone who has to translate a document with special formatting tags that require customization of a OpenTM2 markup table.

Markup tables can also be exported in an SGML-based format for distribution or modification. However, they should only be modified by people experienced in SGML.

Working with OpenTM2 components and functions

General concepts

In this part, you find descriptions of what you can do with all kinds of translation material in

OpenTM2. Each type of data, such as documents, dictionaries, and folders, is considered as an object on which you can perform an action, such as copying, changing, and deleting.
For each **OpenTM2** object, you find a chapter that contains :

- A general description of what the object is
- An overview of the functions and the terminology related to the object
- Detailed descriptions of the actions you can perform on the object

How a task is described

Each task is described according to the following structure:

- In an introduction, the individual task is explained such as when and why to do it.
- How to perform the task is usually described as follows:

Prerequisites

What the prerequisites are

Calling sequence

How to get to the windows required for performing the task

Options and parameters

Which options and parameters to specify and how to specify them

Results

What is achieved by the task

Export and import

To exchange translation material with other **OpenTM2** users or other programs, **OpenTM2** offers functions to exporting and importing the translation material.

To import an object, for example a folder, is a prerequisite for working with it in **OpenTM2**. You export an object to work with it outside **OpenTM2**. Exported objects can be in **OpenTM2** format or in external format.

Objects in **OpenTM2** format are files in a compressed format and can be imported only by other **OpenTM2** users. For files in this format, you only have to specify the drive where to export to, or where to import from. These files are stored in the `eqf\export` directory of the selected drive.

Objects in external format are ASCII encoded files that can be edited with a text editor. They can be imported by other **OpenTM2** users and processed by other systems. For files in external format you must specify the fully qualified DOS file name where to export to, or where to import from. **fully qualified file names**

Files in external format contain SGML tags that describe the structure of the files and the data it contains. For a complete description of SGML, refer to *ISO 8879, Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML)*.

Printing

In several windows you can request to print information.

You can print:

- The contents of all list windows, such as the "Folder List" window or the "Dictionary List" window
- The details of a folder to be imported
- The terms in an exclusion list
- The terms in a new terms list or a found terms list
- The data of a dictionary entry
- A complete dictionary or selected parts of it
- A document file
- The results of counting words
- The key assignments in the Translation Environment

You are prompted with the "Print Setup" window where you can specify the destination of your printout.

Printing list windows

You can print the contents of all list windows, such as the "Folder List" window or the "Dictionary

List" window on a printer of your selection.

Prerequisites

The respective window that contains the list to be printed is active.

Calling sequence

Select **Print list...** from the **File** menu. The "Print Setup" window is displayed where you can specify the printer you want and the layout of your printout.

Results

The list is printed on the printer of your choice. The printout reflects the columns and column widths of the list window that is displayed. If the list window has autotext columns, for example for long document names, the width of the printed columns is adjusted to the actual length.

Windows

To perform an action on an object in **OpenTM2**, you can specify the options and parameters in a window specific for this action. At the bottom of the window you find pushbuttons, where you can:

- Start the action (left pushbutton)
- Leave the window without any further action (**Cancel**)
- Ask for information about this window (**Help**)

Message windows

When you work with **OpenTM2**, the system prompts you with message windows if necessary. There are several types of messages:

Informational

For example, to inform you that a **Translation Memory** was imported completely

Error

For example, if something critical happens

Warning or Question

For example, to ask whether you really want to delete a folder

All messages have a unique number and start with **EQF** to identify them as OpenTM2 messages.

With all message windows where you are asked to confirm actions on an object, you can select a **Yes to All** button in the first message window that pops up after an action has been requested. When you pressed this button, OpenTM2 assumes the answer **Yes** on all subsequent message windows that ask for a confirmation. For example, while analyzing documents of a folder, a message asks if you want to overwrite the target document because translation for a document has already started. If you answer **Yes to All**, all subsequent messages are suppressed.

Progress windows

Some actions may require some time to complete, for example:

- Organizing a dictionary
- Exporting a **Translation Memory**
- Analyzing all documents of a folder

The progress of such an action is indicated by a so-called progress window (see [Figure 19](#)).

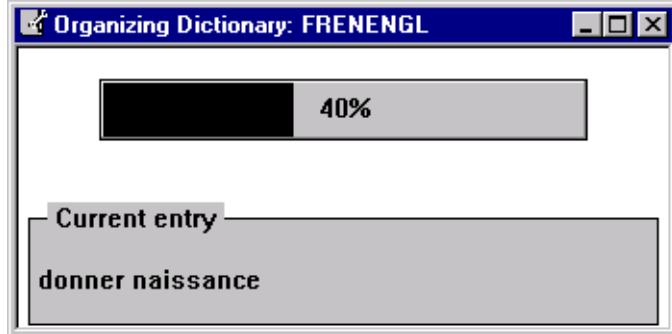


Figure 19. Organizing Dictionary sample window

This type of window can be minimized to let you do other work in **OpenTM2**. When the action is completed, a message with the result of the action is displayed.

If you want to interrupt the action, double-click the system icon of the progress window. In the
Copyright IBM 2010

following message window, you can confirm the cancellation request.

Menus

To perform an action on a list window such as the "Translation Memory List" window in **OpenTM2**, you can do one of the following:

- Select the appropriate item from the OpenTM2 menu bar to access a menu of actions.
- Click the icon in the toolbar representing this action. The most important tasks are represented as icons.
- Open a context-sensitive popup menu where you can select the same items as from the OpenTM2 menu bar. The menu shows only the selection of tasks that can be started from the currently active list window. You open the popup menu by pressing the right mouse button.

Specifying an output file

You may need to specify an output file when working with one of the following windows:

- "Count Words" for documents or a folder (see [Figure 19](#))
- "Print Dictionary" (see [Figure 20](#))

When you select the output option **File** and click **Select...**, the "Select File" window is displayed (see [Figure 20](#)).

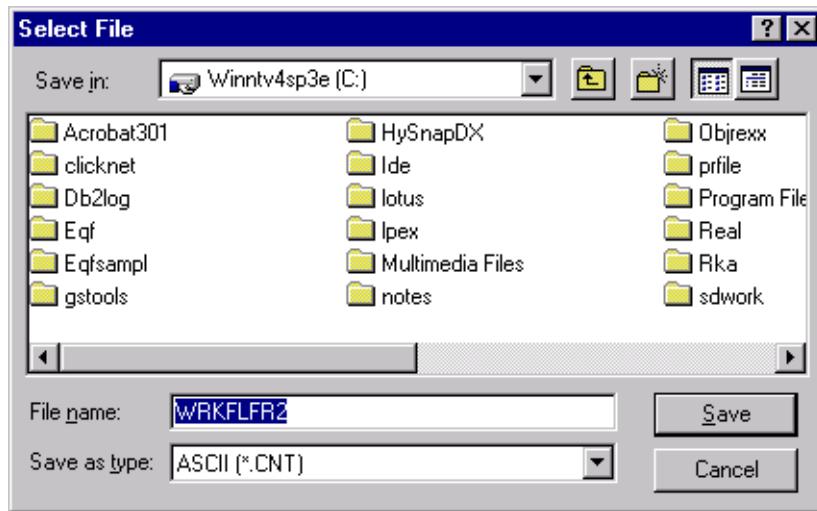


Figure 20. Select File window

Specify the fully qualified file name, consisting of drive, directory and file name, as follows:

1. Select the drive where the file is to reside from **Save in**.
2. Select the directory to hold the file by double-clicking it.
3. Specify a name in the **File name** field.
4. Select the file type from **Save as type**.

Alternatively, you might select an existing file name as the output file. Its content will be overwritten upon completion of the operation.

When you have specified the file name, click **Save** to return to the previous window.

Results

The drive, directory, and file name you select are used by the respective function.

Selecting multiple objects in list windows

Several list windows in OpenTM2 allow you to select more than one object on which you want to perform the same task. This allows you to perform repeatable and time-consuming tasks unattended.

In the following chapters, whenever a "Calling sequence" asks you to select "one or more" objects in one of the list windows, you can use this capability. If unsure, just try to select multiple objects in a window. OpenTM2 only allows you to select multiple objects if it can safely perform the task on multiple objects in one run.

The selection of multiple objects is done much like in the Windows environment. You use the mouse to click list items. Holding down the Ctrl key, and then clicking list items, lets you randomly select list items in a list window. Holding down the Alt key, and then clicking list items, lets you

select a series of list items. If you prefer to use key combinations only, or need information about more sophisticated selection methods, see the Windows Help facility.

Multiple object selection is supported in the following windows:

Document List window

Document List windowselecting multiple documents Almost all OpenTM2 tasks can be applied to multiple documents in this window.

Note:

If you view document properties for multiple documents, the "Document Properties" window shows:

- Property values that are equal for all selected documents in black color.
- Property values that are different for all selected documents in grey color. Only the property values of the first document are shown.

If you change document properties for multiple documents, the changed values are applied to all selected documents, and unchanged values are left as they are.

For example, assume that document A (properties are markup EQFAMRI, and editor STANDARD) and document B (properties are markup EQFAMRI, and editor XLATE) are selected in the "Document List" window. The "Document Properties" window shows markup EQFAMRI in black color (because it is equal for both documents) and editor STANDARD in grey (because the second document uses a different editor). If you change the editor in the "Document Properties" window to RTFEDIT, and save the change, this editor is now associated with the selected documents.

Folder List window

Folder List windowselecting multiple folders The Export, Delete, and Analyze tasks can be applied to multiple folders in this window.

Note: If you prefer to use the Explorer-like tree view, you cannot select multiple folders from the directory tree displayed on the left. Instead, click the parent folder on the left, and click the individual folders on the right.

Translation Memory List window

Translation Memory List windowselecting multiple Translation Memories The Export, Delete, and Organize tasks can be applied to multiple Translation Memories in this window.

Dictionary List window

Dictionary List windowselecting multiple dictionaries The Export, Delete, and Organize tasks can be applied to multiple dictionaries in this window.

Connect Shared Translation Memories window

Connect Shared Translation Memories windowselecting multiple shared Translation Memories The Access and Disconnect tasks can be applied to multiple shared Translation Memories in this window.

Connect Shared Dictionaries window

Connect Shared Dictionaries windowselecting multiple shared dictionaries The Access and Disconnect tasks can be applied to multiple shared dictionaries in this window.

Import ... windows

Import Documents windowselecting multiple documents **Import Folder window**selecting multiple folders **Import Translation Memory window**selecting multiple Translation Memories **Import Dictionary window**selecting multiple dictionaries **Import New Terms window**selecting multiple new terms lists **Import Found Terms window**selecting multiple found terms lists **Import Exclusion Lists window**selecting multiple exclusion lists **Import Markup Table List window**selecting **Markup Table List** Import tasks can be applied to multiple objects in the various import windows, such as the "Import Documents", "Folder", "Translation Memory", "Dictionary", "New Terms Lists", "Found Terms Lists", "Exclusion Lists", and "Markup Table" windows.

Notes:

- In the "Import Folder" window the options you specify are applied to all folders to be imported. The option cannot be selected if multiple folders are to be imported. Also, the "Details" page in the "Import Folder" window cannot be selected.
- In the "Import Dictionary" window the option cannot be selected if multiple dictionaries are to be

imported. The dictionaries are imported using the file names given when they were exported. For every dictionary to be imported the "New Dictionary" window is shown before the import starts.

- In the "Import Translation Memory" window the option cannot be selected if multiple Translation Memories are to be imported. The Translation Memories are imported using the file names given when they were exported. For every Translation Memory to be imported the "New Translation Memory" window is shown before the import starts.

Viewing and changing the details of a list item

In the list windows, **OpenTM2** can show you more information in addition to the names of the list items. You can ask the system to display these view details selectively and in a sequence defined by you.

The details contain properties and statistical information of the list items, such as the:

- Name of a document
- Source language of a **Translation Memory**
- Size of a new terms list
- Date when a document was analyzed
- Date when a found terms list was updated
- Drive where a folder is stored

Details can be displayed for:

- Folders
- Documents
- Dictionaries
- **Translation Memory databases**
- New terms lists
- Found terms lists
- Exclusion lists
- Markup table list

You can view the selected details and change which details are displayed on a list window.

Viewing selected details

Prerequisites

The respective list window is already selected.

Calling sequence

Select **Details** from the **View** menu.

Results

The list window contains for each list item the detail information as previously defined. As an example, the "Document List" window is shown in [Figure 21](#).

Name	Compl.%	Size	Imported	Analyzed
DEVICE.HTML	0	1393	29.03.2000	29.03.2000
itmtesto.txt	0	195	17.11.2000	
tm\project\part01...	0	4608	25.10.2000	
tm\project\part02...	0	4608	25.10.2000	
tm\project\part03...	0	4608	25.10.2000	
tm\project\part04...	0	4608	25.10.2000	
TRANS.HTML	0	1280	29.03.2000	

Figure 21. Document List window sample with selected details

[PIC]Figure shows Document List window sample with selected details

As a default setting, only the names are displayed. You can change the details to be displayed (see [Changing the details setting](#)). Once you have defined your setting of the details, they are displayed as selected.

Changing the details setting

OpenTM2 lets you customize the view details in the respective list window.

If not all details of a list item, for example a folder, fit on one line in the list window, you should set priorities.

As an example, this window is explained for the details of documents in the "Document List" window.

Prerequisites

The respective list window is already selected, for example a "Document List" window.

Calling sequence

Select **Change Details...** from the **View** menu.

The "Change View Details" window is displayed (see [Figure 22](#)).

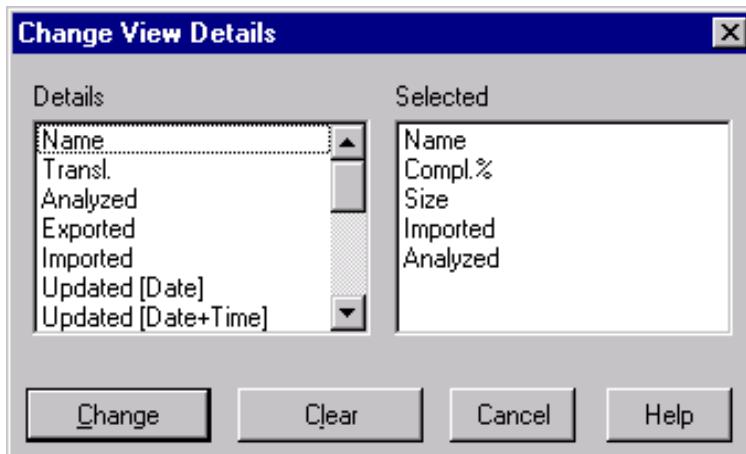


Figure 22. Change View Details sample window for documents

Options and parameters

Details

This box displays all the view details available, for example, name, drive, dates. Double-click those view details you want to be displayed in the "Document List" window.

Selected

This box displays the view details selected. The items are displayed from left to right in the same order you selected them from the **Details** list box. To deselect an item, double-click the entry in the **Selected** box. Click **Clear** to remove all items from the **Selected** box. You can then begin to define a new sequence of details.

When you have selected the desired details, click **Change** to activate the new setting.

Click **Cancel** at any time to leave the "Change View Details" window. No changes are made.

Results

The list window contains the selected details in the newly defined sequence. This setting remains active throughout the current session until you change it again.

To save the settings for future **OpenTM2** sessions, respond with **Yes** to the prompt *Do you want to save the current workbench?* when you leave **OpenTM2**. The current settings will then be used the next time you start **OpenTM2**.

Arranging and filtering list items

In the list windows, all items are listed and sorted by name in alphabetical order. For example, in the "Folder List" window, you always find all the folders that you created in, or imported into, **OpenTM2**, sorted by name.

You can change this by arranging the items according to specific details and by filtering specific items. For example, you can arrange the folders by name in descending order, or by markup table and within the same markup table by description. In addition, you can define that only those folders are displayed whose name starts with a **W**.

You can arrange and filter items in the following list windows:

- "Folder List"

- "Document List"
- "Dictionary List"
- "Translation Memory List"
- "New Terms Lists"
- "Found Terms Lists"
- "Exclusion Lists"
- Markup table list

Arranging list items

list windowsarranging list items OpenTM2 provides several sort keys that enable you to arrange the items in the list windows according to your needs.

Prerequisites

The respective list window is selected.

Calling sequence

Select **Sort...** from the **View** menu.

The "Sort List Items" window is displayed (see [Figure 23](#)).

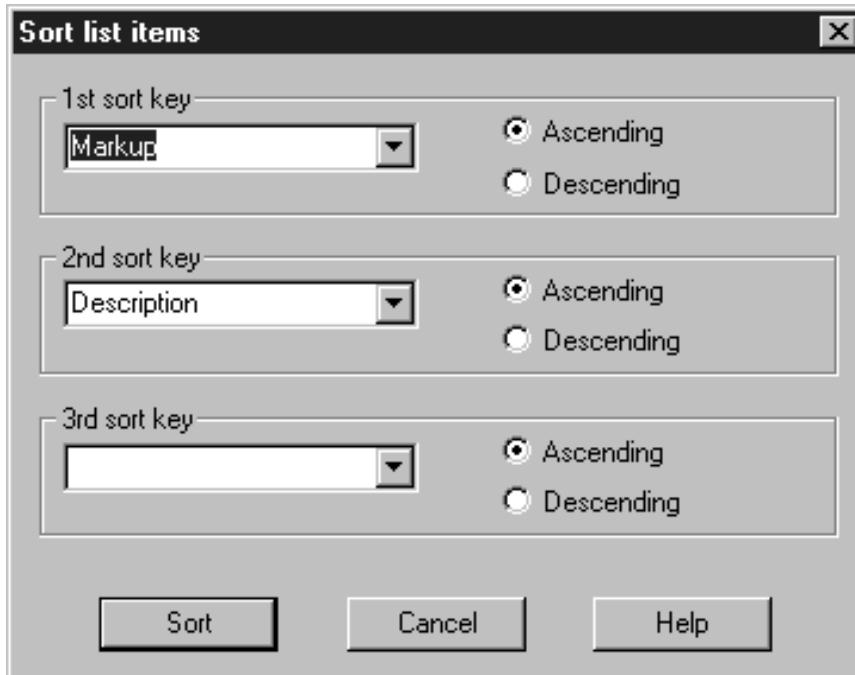


Figure 23. Sort List Items window
[\[PIC\]](#)Figure shows Sort List Items window

Options and parameters

1st sort key

Select the detail according to which the list is to be sorted, for example, markup table. Also decide whether items are to be sorted in ascending or descending order.

2nd sort key

Select the detail according to which the list is to be sorted within the first sort and decide whether the items are to be sorted in ascending or descending order. For example, when you decide to sort your folders by markup table (1st sort key) and several folders have the same markup table, these folders are automatically sorted by name within the markup table. If, however, you want to sort the folders that use the same markup table by their description, you select **Description** as the 2nd sort key.

3rd sort key

Select the detail to be used as third sort key and specify the direction of the sort.

After selecting your sort keys, click **Sort** to reflect your specifications in the appropriate list window.

Click **Cancel** to leave the "Sort List Items" window without rearranging the items on your list window.

Results

The list window displays the items sorted according to your specifications.

Filtering list items

list windowsfiltering list items If a list window contains too many items, you can reduce its contents.

Prerequisites

The respective list window is selected.

Calling sequence

Select **Filter...** from the **View** menu.

The "Filter List Items" window is displayed (see [Figure 24](#)).

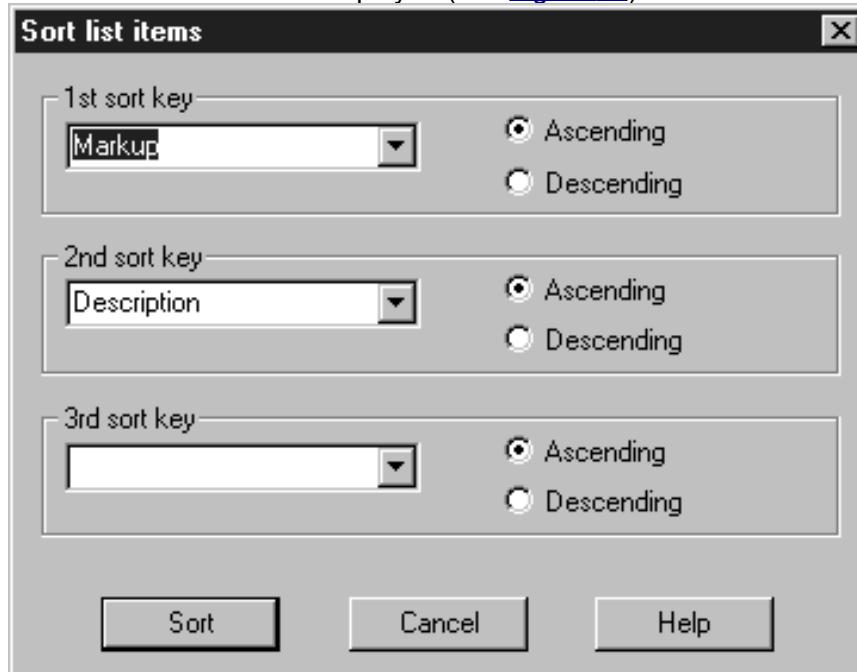


Figure 24. Filter List Items window

[PIC]Figure shows Filter List Items window

Options and parameters

Column

Select the detail that you want to use as filter condition. For example, if you wanted to list only those folders that were updated before, on, or after a certain date, you would select the detail **Updated**.

Operator

Select one of the available operators.

Value

Specify a name or date, or part of a name. It can be up to 15 characters long and be specified in uppercase, lowercase, or mixed case. You can also use a wildcard (*) in the value.

The individual operators and possible values are explained using the following folder list:

Name	Description	Updated	TransMem	Dictionaries
SAMPLAMI	Folder for Sample Ami Pro	22.03.1996	SAMPLAMI	SAMPLAMI
SAMPLEHTML1	Folder for Sample HTML Session 1	29.03.2000	SAMPLEHTML1	SAMPLE1
SAMPLEHTML3	Folder for Sample HTML Session 3	29.03.2000	SAMPLEHTML3	SAMPLE3
SHOWME		27.02.1998	SHOWMEM	SHOWDICT
WORKFLFR	Workbook exercise folder Eng -> Fre	22.03.1996	WORKMFR1	WORKDIR
WORKFLGE	Workbook exercise folder Eng -> Ger	22.03.1996	WORKMGE1	WORKDIGE
WORKFLIT	Workbook exercise folder Eng -> Ita	22.03.1996	WORKMIT1	WORKDIIT

[PIC]Figure shows Folder List window

The Equal operator

Use this operator to search for a single folder that exactly matches a specific value. You cannot use a wildcard in the value.

Example 1: To display only the folder SHOWME, you must specify the following:

Column	Operator	Value
Name	Equal	showme

Figure 25. Specifying the Equal operator (Example 1)

[PIC]Figure shows Specifying the Equal operator (Example 1)

Example 2: To display the folder that was updated on February 27, 1998, you must specify the following:

Column	Operator	Value
Updated	Equal	27.02.1998

Figure 26. Specifying the Equal operator (Example 2)

[PIC]Figure shows Specifying the Equal operator (Example 2)

The Not equal operator

Use this operator to search for all folders that do not match a specific value. You cannot use a wildcard in the value.

Example 1: If you want all folders whose name is *not* SHOWME, your specification will look similar to the one in [Figure 25](#) except that you select the **Not equal** operator instead of the **Equal** operator.

Example 2: If you want all folders that were *not* updated on February 27, 1998, your specification will look similar to the one in [Figure 26](#) except that you select the **Not equal** operator instead of the **Equal** operator.

The Less operator

Use this operator to search for all folders updated before a certain date or for all folders the detail of which precedes the letters that you specify. You can use a wildcard in the value.

Example 1: To list all folders that were updated before March 29, 2000, you specify the following:

Column	Operator	Value
Updated	Less	29.03.2000

Figure 27. Specifying the Less operator (Example 1)

[PIC]Figure shows Specifying the Less operator (Example 1)

Example 2: To list all folders whose name precedes the letters **WORKFLG**, you specify the following:

Column	Operator	Value
Name	Less	WORKFLG

Figure 28. Specifying the Less operator (Example 2)
[\[PIC\]](#)Figure shows Specifying the Less operator (Example 2)

You then get the following list:

Folder List [Some]				
Name	Description	Updated	TransMem	Dictionaries
SAMPLAMI	Folder for Sample Ami Pro	22.03.1996	SAMPLAMI	SAMPLAMI
SAMPLEHTML1	Folder for Sample HTML Session 1	29.03.2000	SAMPLE...	SAMPLE1
SAMPLEHTML3	Folder for Sample HTML Session 3	29.03.2000	SAMPLE...	SAMPLE3
SHOWME		27.02.1998	SHOWMEM	SHOWDICT
WORKFLFR	Workbook exercise folder Eng -> Fr	22.03.1996	WORKMF...	WORKDIFR

Figure 29. Result of a Less operation
[\[PIC\]](#)Figure shows Result of a Less operation

The Less or equal operator

Use this operator to search for all folders updated on or before a certain date or for all folders the detail of which precedes or matches the letters that you specify. If you specify a wildcard in the value or do not specify the full value, you will not get an exact match.

Example 1: To list all folders that were updated on or before February 27, 1998, you specify the following:

Column	Operator	Value
Updated	Less or equal	29.03.2000

Figure 30. Specifying the Less or equal operator (Example 1)
[\[PIC\]](#)Figure shows Specifying the Less or equal operator (Example 1)

You then get all folders updated in 1996, and the folder updated on the specified date.

Example 2: To list all folders whose name precedes or matches the letters **WORKFLG**, you specify the following:

Column	Operator	Value
Name	Less or equal	WORKFLG

Figure 31. Specifying the Less or equal operator (Example 2)
[\[PIC\]](#)Figure shows Specifying the Less or equal operator (Example 2)

In this example you get the same list as in [Figure 6](#) because the value **WORKFLG** is not the full name of a folder and therefore no exact match can be listed.

The Greater operator

Use this operator to search for all folders updated after a certain date or for all folders the detail of which follows the letters that you specify. You can use a wildcard in the value.

Example 1: To list all folders that were updated after March 22, 1996, you specify the following:

Column	Operator	Value
Name	Greater	22.03.1996

Figure 32. Specifying the Greater operator (Example 1)
[\[PIC\]](#)Figure shows Specifying the Greater operator (Example 1)

Example 2: To list all folders whose dictionary name follows the letters **WORKDIF**, you specify the following:

Column	Operator	Value
Dictionaries	Greater	WORKDIF

Figure 33. Specifying the Greater operator (Example 2)
[\[PIC\]](#)Figure shows Specifying the Greater operator (Example 2)

You then get the following folders:

Folder List [Some]				
Name	Description	Updated	TransMem	Dictionaries
WORKFLFR	Workbook exercise folder Eng -> Fre	22.03.1996	WORKMFR1	WORKDIFR
WORKFLGE	Workbook exercise folder Eng -> Ger	22.03.1996	WORKMGE1	WORKDIGE
WORKFLIT	Workbook exercise folder Eng -> Ita	22.03.1996	WORKMIT1	WORKDIIT

Figure 34. Result of a Greater operation
[\[PIC\]](#)Figure shows Result of a Greater operation

The Greater or equal operator

Use this operator to search for all folders updated on or after a certain date or all folders the detail of which follows or matches the letters that you specify. If you specify a wildcard in the value or do not specify the full value, you will not get an exact match.

Example 1: To list all folders that were updated on or after February 27, 1998, you specify the following:

Column	Operator	Value
Updated	Greater or equal	27.02.1998

Figure 35. Specifying the Greater or equal operator (Example 1)
[\[PIC\]](#)Figure shows Specifying the Greater or equal operator (Example 1)

You then get all folders updated in 2000, and the folder updated on the specified date.

Example 2: To list all folders whose **Translation Memory** name follows or matches the letters **SH***, you specify the following:

Column	Operator	Value
TransMem	Greater or equal	SH*

Figure 36. Specifying the Greater or equal operator (Example 2)
[\[PIC\]](#)Figure shows Specifying the Greater or equal operator (Example 2)

In this example you get the following list:

Name	Description	Updated	TransMem	Dictionaries
SHOWME		27.02.1998	SHOWMEM	SHOWDICT
WORKFLFR	Workbook exercise folder Eng -> Fre	22.03.1996	WORKMFR1	WORKDIFR
WORKFLGE	Workbook exercise folder Eng -> Ger	22.03.1996	WORKMGE1	WORKDIGE
WORKFLIT	Workbook exercise folder Eng -> Ita	22.03.1996	WORKMIT1	WORKDIIT

Figure 37. Result of a Greater or equal operation
[\[PIC\]](#)Figure shows Result of a Greater or equal operation

The Like operator

Use this operator to search for folders the detail of which matches, starts with, or ends with, the value that you specify. If you do not specify the full name, you must represent the missing characters by an asterisk. For example, if you want to list only those folders whose description starts with **Folder**, you specify the following:

Column	Operator	Value
Description	Like	Folder*

Figure 38. Specifying the Like operator
[\[PIC\]](#)Figure shows Specifying the Like operator

After making your selections, click **Activate filter** to reflect them in the appropriate list window.

Click **Cancel** to leave the "Filter List Items" window without changing the contents of your list window.

If you want to have a complete list of your folders again, select **Show all** from the **View** menu.

Viewing and changing the system preferences

properties system preferences system preferences, viewing and changing You can view and change the properties of your system.

Calling sequence

Select **System Preferences** from the **File** menu.

The "General page" of the "System Preferences" window is displayed. For more advanced options select the "Advanced" page of this window.

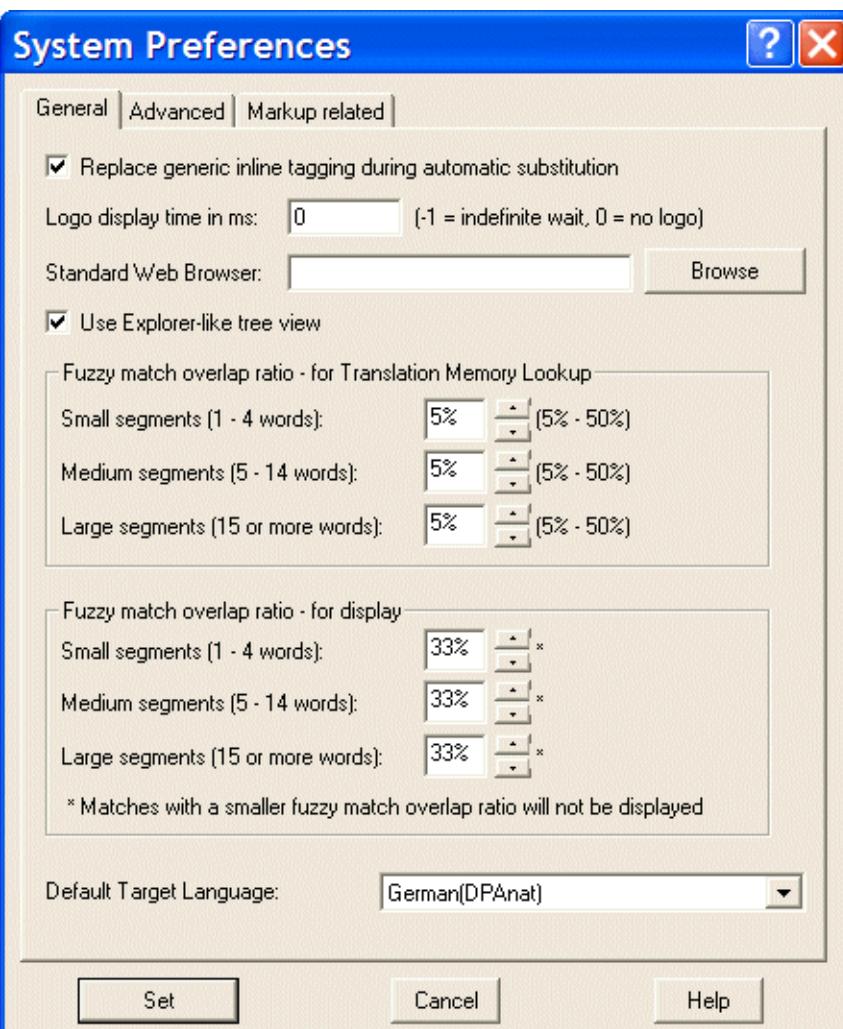


Figure 39. General page of System Preferences window

Options and parameters

Replace generic inline tagging during automatic substitution

replace generic inline tagging inline tagging, replace Select this check box if you want OpenTM2 to adapt the tagging in the Translation Memory during automatic substitution for exactly matching segments that differ only in inline tags. If selected, OpenTM2 replaces the tagging in a translation proposal and the source of this proposal with the tagging from the currently active segment. As a result, the Translation Memory becomes format-independent, which results in more automatic translations because of more exact matches. **Translation Memoryformat-independence**

The automatic tagging replacement occurs only if:

- Exact text matches are encountered
- Tagging between the currently active segment and the translation proposal is different
- A one-to-one correspondence between the taggings in the currently active segment, the translation proposal, and the source of the proposal can be proven.
If an active segment does not contain a tagging that is, on the other hand, contained in a translation proposal, OpenTM2 deletes this tagging from the translation proposal and the source of this proposal.
- All taggings in a segment are suitable for replacement or deletion. No partial replacement or deletion takes place.

This option is helpful if you translate updated documents that were meanwhile converted to a different document format. For example, if you first translated a BookMaster document, and you now translate an update of the same document in SGML format, the Translation Memory for this document successively converts to SGML.

As a side effect, inline tags convert to lower, upper, or mixed case, dependent on the preference in the translated document.

Logo display time in ms

Specify for how many milliseconds (ms) the **OpenTM2** logo is to be displayed when you log on to **OpenTM2** before the Translation Workbench appears.

Standard Web Browser

Specify the name of the Web Browser to be called. You can also click **Browse** to display the "Select a Web Browser" window on which you can select it.

Use Explorer-like tree view

Select this check box if you want to have your folder list displayed in an Explorer-like tree view. Your "Folder List" window would then look as follows:

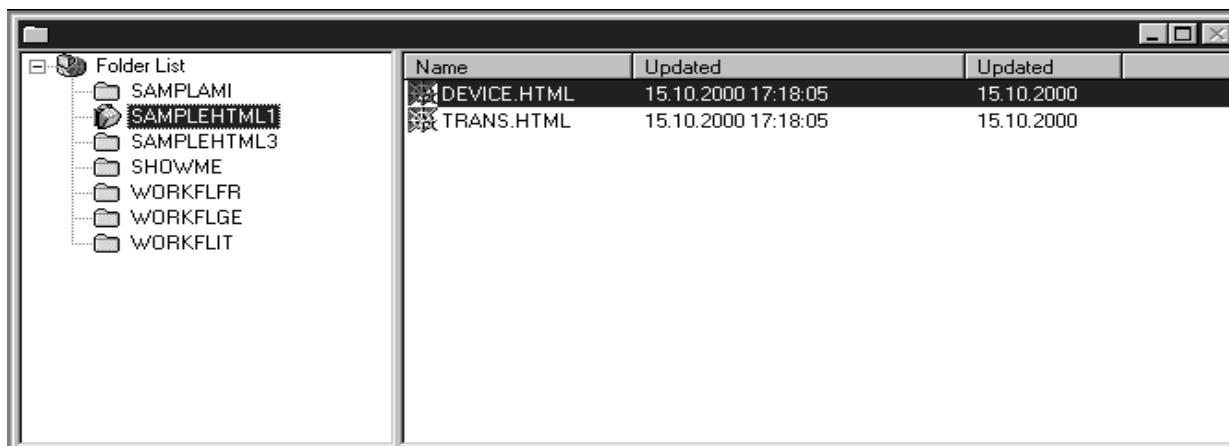


Figure 40. Folder List window in Explorer-like tree view

[PIC]Figure shows Folder List window in Explorer-like tree view

Fuzzy match overlap ratio — for Translation Memory Lookup

fuzzy match overlap ratio — for translation memory lookupdefault individual settings You can specify individual overlap ratio thresholds for the retrieval of fuzzy matches in small, medium, and large segments. You can specify values from 5% to 50%. If nothing is specified, the default overlap ratio is 33% regardless of a segment's length.

Fuzzy matches with an overlap smaller than the specified percentage (or default) are not retrieved from the translation memory as translation proposals. Further, all matches below 50% are

considered as "no matches" when OpenTM2 counts matches (see [Counting the number and type of matches in a document](#) and [Creating a Counting Report](#)).

See [Translation Memory matches](#) for more details, if required.

Fuzzy match overlap ratio - for display

For the display in the translation environment, you can specify individual overlap ratio thresholds for fuzzy matches in small, medium, and large segments. You can specify values from 5% to 99%. If nothing is specified, the default overlap ratio is 33% regardless of a segment's length.

Fuzzy matches with an overlap smaller than the specified percentage (or default) are not displayed as translation proposals in the translation environment.

Default Target Language:

For the editor and translation memory handling as well as some other functionalities, it is necessary to specify the default target language.

Memory import in ANSI or ASCII format:

- The default target language is used to obtain the codepage which is used to convert the input data from ASCII/ANSI to Unicode (UTF-16). It is therefore important that the "default target language" matches exactly the language of the target segments in the imported memory.

Memory export in ANSI or ASCII format:

- The default target language is used to obtain the codepage which is used to convert the internal Unicode (UTF-16) data to ASCII/ANSI. When the default target language does not match the target language of the exported segments most likely some special characters will be corrupted. Memories with a source language other than English should always be exported in Unicode format to avoid corrupted characters on the source or the target side of the segments.

Other:

- There are several other locations within OpenTM2 which require a conversion to/from Unicode. At all locations where no specific language is available (i.e. all non-document related functions) the default target language is used to obtain the codepage for the conversion.

After making your selections, click **Set** to activate the new settings.

Click **Cancel** to leave the "System Preferences" window without activating any new settings.

Results

The settings become effective immediately. If you selected the Explorer-like tree view for your folder list, however, close **OpenTM2** and then open it again to get the "Folder List" window in this view.

Working with documents

You can translate many document types with **OpenTM2**, such as letters, brochures, booklets, and manuals.

You can translate documents that were originally created with a variety of text editors, word processors, and publishing systems. Document formats you can use with **OpenTM2**. The file types you can use with **OpenTM2** include:

- ASCII text files
- Documents from word processors and publishing systems:
 - Ami Pro
 - BookMaster^(R)
 - and IPF
 - HTML
 - Microsoft^(R)
 - Word for
 - Windows^(R)
 - RTF
 - WordPerfect
 - SGML
 - FrameMaker
 - Interleaf
 - Lotus^(R)
 - Notes^(TM)
 - Ventura
 - PageMaker
 - QuarkXpress

- Machine-readable information (MRI)
- Microsoft®
- PowerPoint®
- Machine-readable information that is contained within:
- Assembler source code
- Help files of Windows® programs
- Help and message files of OS/2® programs
- Windows® resource files
- OS/2® resource files
- Unicode documents

A translation project can consist of one file or a set of files. In **OpenTM2**, each file is treated as a document.

In their original formats, documents consist of text and layout information, called *markup*. Markup defines things such as the positioning of page breaks and paragraph endings, highlighting, and font sizes.

layout information in a document
formatting information in a document markup
language description

When translating a document, **OpenTM2** must be able to distinguish between this markup and the text that is to be translated. By ignoring the markup information during translation, **OpenTM2** preserves the original format of the document—although minor changes in the translated document can be necessary, such as adding space to a table that is full or a translated term that does not fit within a column. **OpenTM2** uses a different *markup table* for each document format type.

Each document has a number of *properties*, such as the source and target languages, the markup table to use with the document, and the name of the **Translation Memory** to use for translations.

You must specify the properties of a document when you import it into **OpenTM2**.

By default, all documents in a folder have the same properties, although you can change them for individual documents.

For further information on how to deal with document formats, see [Dealing with specific document formats](#).

Overview and terminology

You must first *import* documents you want to translate into **OpenTM2** (see [Importing a document](#)).

This can be done by importing documents into an existing folder. At this point, **OpenTM2** asks you to specify the properties of the document. You can either let the system assign the default properties of the folder to which the document belongs, or you can assign different properties to the document.

Another method for importing documents is to import a complete folder that contains at least one document (see [Importing a folder](#)).

When you go to the **Translation Environment** to begin translating a new document, **OpenTM2** analyzes the document to distinguish between the text that requires translation and markup data. It then splits up the document into translatable *segments*—which usually correspond to individual sentences—and markup data that is not to be translated.

OpenTM2 can distinguish between abbreviations ended by a period and the end of a sentence. It uses an abbreviation list to correctly identify sentence boundaries. See [Maintaining abbreviation lists](#) to learn how to maintain an abbreviation list.

The **OpenTM2** editor displays the segments to be translated and the markup in different colors so that you can easily distinguish between the two. As you move from segment to segment, the markup data is skipped.

With the **OpenTM2** editor, you can check the spelling in the translated document and see suggestions for correct spellings.

OpenTM2 also includes a *word count* function, which shows you the number of words and markup tags in the original and translated documents.

Prior to translating a document, you can check a document against the dictionaries that are installed with **OpenTM2** to compile a list of terms in the document that are not currently in the dictionaries. You can choose to add terms from this *new terms list* to a dictionary.

When translating a new version of an existing document that has already been translated, you should check whether a **Translation Memory** for the document already exists, or whether one can be generated.

To continue translating a document that has been partly translated by somebody else, you should first acquire all related translation material. In particular, ask for the **Translation Memory** and all the dictionaries that were used. These can be passed to you in the form of an exported folder.

You can then display details of the document in the "Document List" window, such as the properties and processing status of a document, the date of import, date of analysis, and what percentage of the document has already been translated.

After completing a translation, you can choose to **export** a document back into its original format to load it into the original word processing system for a final layout review.

What you can do with documents

OpenTM2 offers several processing functions for documents, which you start by selecting:

1. At least one document from the "Document List" window
2. An action from the associated **File**, **View**, or **Utilities** menu

Prerequisites

A folder containing at least one document (except if you import a document).

Calling sequence

Select:

1. The folder with your documents from the "Folder List" window
2. **Open** from the **File** menu

Alternatively, select the "Folder List" window and double-click the folder name in the list. The "Document List" window for this folder is displayed.

Results

The selected folder is "opened". To perform a task on a document, select at least one document and an action from one of the toolbar menus.

Analyzing a document

Analyzing a document means splitting it up into translatable pieces called *segments*.

Segments are in most cases equivalent to sentences. This segmentation is performed with the help of *markup tables* that contain specific information for the type of markup used in the document. You choose the markup table to use as one of the properties of the document.

After the analysis of a document, **OpenTM2** produces a copy of the original document to be used for the translation, this means, the original version of the document is kept unchanged by **OpenTM2**.

The system allows you to analyze documents with different results:

- The documents are segmented (default).
- The documents are segmented, segments can be substituted with segment translations, or a **Translation Memory** can be filled with these translations.
- The documents are segmented, and terminology lists are created.

To analyze all documents in a folder, see [Analyzing all the documents in a folder](#).

Analyzing documents using defaults

Segmentation is the basic part of the analysis. You activate it implicitly when you select **Open** for a document that has not yet been analyzed. **document opening**

Prerequisites

The folder containing the document must be opened.

Calling sequence

Select:

1. The document to analyze from the "Document List" window of the folder
2. **Open** from the **File** menu

Alternatively, double-click the document in the "Document List" window.

This takes you to the **Translation Environment** where the analyzed document is displayed in the "Translation" window.

Results

The document is segmented. You can start to translate it (see [Translating a document](#)).

Analyzing documents using Translation Memory databases

analyzingdocuments using Translation Memory databases If you have **Translation Memory databases** available from previous translations, you have several options:

- **OpenTM2** can search existing **Translation Memory databases** for segments that match segments in the document to be analyzed. Segments that match exactly or almost exactly can be copied into a new document- or project-specific **Translation Memory**.
- The system can perform automatic substitutions, that is, it copies the translations of segments that were previously made for an identical source segment (*exact match*) into the translation version of the document. Matches are considered to be exact even if the tagging of the segment to be replaced differs from that of the translation proposal.
- The remaining untranslated segments can optionally be send to the machine translation server (only if the target language is supported by the machine translation server).

Prerequisites

- The **Translation Memory databases** to be searched must exist.
- The folder containing the document must be opened.

Calling sequence

Select:

1. One or more documents from the "Document List" window
2. **Analyze** from the **File** menu

The "Analyze Documents" window (see [Figure 44](#)) is displayed.

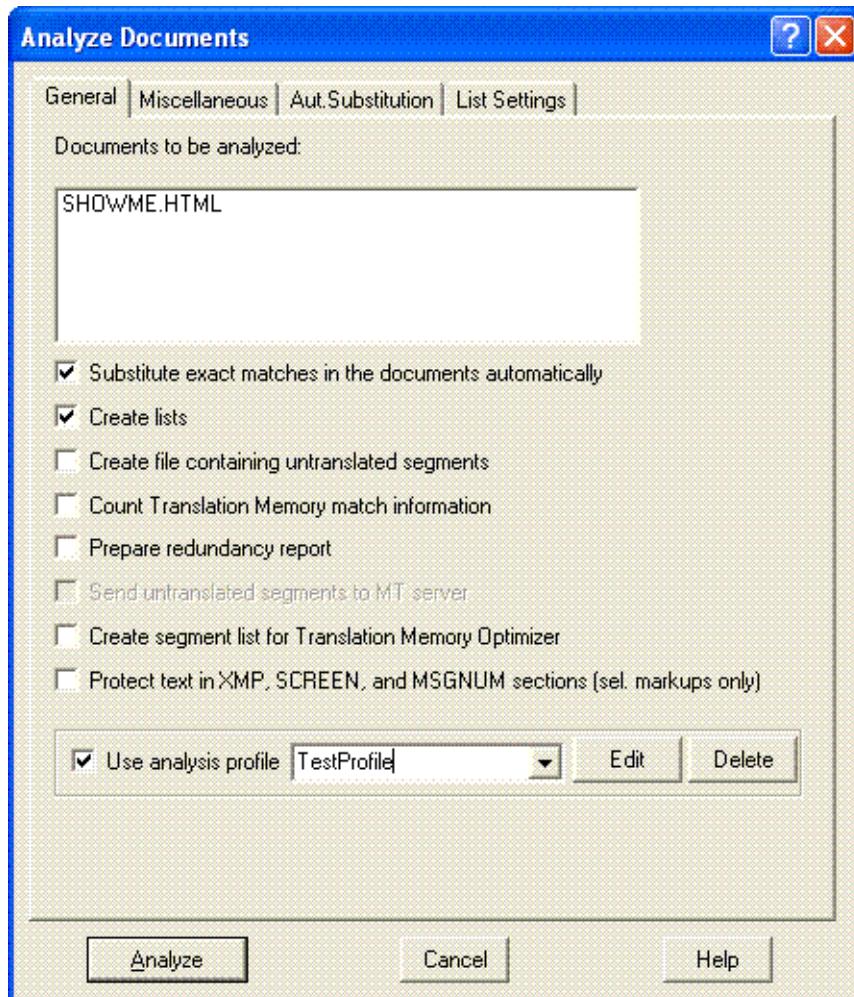


Figure 44. Analyze Documents window

Options and parameters

On the "General" page:

Documents to be analyzed

This box contains the documents you selected previously.

Substitute exact matches in the documents automatically

Select this option to automatically replace original segments with the translations found in the **Translation Memory** associated with the document. The document is then already partially translated when you start working with it, enabling you to concentrate on the translation of new segments.

This option is particularly useful in conjunction with the **Add segments to Translation Memory of folder/document** option on the "Miscellaneous" tabbed page. Many **Translation Memory databases** can be searched in one run for matching segments that are copied into the document-specific **Translation Memory**, and are substituted in the translation document.

When you select this option, the **Aut. Substitution** tab is created, which is described in the following.

Create lists

When you select this option, the **List Settings** tab is created, which is described in [Analyzing documents using dictionaries](#).

Create file containing untranslated segments

untranslated segmentsprocessing them untranslated segmentscollecting them in a file
untranslated segmentssending them to a machine translation system machine translation
systemSelect this option if you want OpenTM2 to create a separate file containing all untranslated segments. If you also selected **Substitute exact matches in the documents automatically**, this file contains only the remaining untranslated segments.

A file with a format similar to an external Translation Memory is created (see [Untranslated segments file](#)) containing empty target segments. This file can be exported for use by other systems . If an external system is able to add valid translations for these untranslated segments, the updated file can be imported into OpenTM2 as a **Translation Memory** for translation use.

Count Translation Memory match information

Select this option to enable OpenTM2 to count the number and type of matches that are found in the **Translation Memory** for the document. The document is then segmented and the matches are counted but no segments are replaced with matches that are found in the **Translation Memory**.

If you also select the **Substitute exact matches in the documents automatically** option, OpenTM2 automatically replaces segments with matches found in the **Translation Memory** and counts the number and type of matches replaced.

For information on how to view the counting result, refer to [Counting the number and type of matches in a document](#).

Prepare Redundancy Report

Select this option if you later want to create a Redundancy Report to calculate the costs for a translation project. For more information on Redundancy Reports, refer to [Creating a Redundancy Report](#).

Send untranslated segments to MT server

Select this option if you want send untranslated segments to the MT (Machine Translation) server. This option is inactive in the current version of OpenTM2.

Create segment list for Translation Memory Optimizer

Select this option if you want to create a segment list for the Translation Memory Optimizer too

Protect text in XMP and SCREEN sections

Select this option to protect text inside of <xmp> and <screen> sections in IDDOC or DITA documents. Without this option the text will be translatable.

Use analysis profile

Select this option to use an analysis profile for the analysis of documents. An analysis profile contains markup table specific analysis settings which will override the analysis settings specified in the analysis window. Select an analysis profile from the analysis profile list or enter the name of a new analysis profile. Press the **Edit** button to specify or modify the analysis settings in the profile. Press the **Delete** button to delete the currently selected analysis profile.

On the "Miscellaneous" page:

Add segments to Translation Memory of folder/document

Select this option if you want the Translation Memory databases that you can select from the **Available** list to be searched for matches. All exact matches or fuzzy matches found in the selected Translation Memory databases are then added to the Translation Memory databases associated with the folder or documents that are being analyzed.

From the **Available** list, select with a double-click those **Translation Memory databases** from where the segments are to be copied. The selected **Translation Memory databases** are displayed in the **Selected** box. You must select at least one **Translation Memory**, and you can select a maximum of 10 **Translation Memory databases**.

Stop search at first memory with exact match

Select this option if the search for memory segments should stop at the first memory containing an exact match. This avoids duplicate exact matches from subsequent Translation Memory databases in the list. Without this option OpenTM2 will extract matches from all selected Translation Memory databases.

On the "Aut. Substitution" page, which is only shown when the **Substitute exact matches in the documents automatically** option is selected:

Use exact context match when more than one found

Select this option if you expect segments with more than one exact match in the associated **Translation Memory** and you want the system to choose the one from the same document. In this case, the document must contain only one exact match. If this option is not selected, OpenTM2 skips the automatic substitution for a segment when more than one exact translation is found for a source segment.

Use latest exact match when more than one is found

Select this option if you expect segments with more than one exact match in the associated **Translation Memory** and you want the system to choose the most recent translation of the segment. If this option is not selected, OpenTM2 skips the automatic substitution for a segment when more than one exact translation is found for a source segment.

Adjust leading white space to white space in source segment

Select this option if the text is to have the same leading white space (blank, tab stop, and line feed) before and after the automatic substitution.

Adjust trailing white space to white space in source segment

Select this option if the text is to have the same trailing white space (blank, tab stop, and line feed) before and after the automatic substitution.

Respect CRLF in segment

The Translation Memory may contain two matching target segments for a source segment, one with carriage return line feed (CRLF), and one without. If you select this option, the target segment containing CRLF is taken during automatic substitution.

No blank at segment end even if source segment has one

A source segment may differ from its matching target segment in that the source segment contains one or more blanks at its end. If you select this option, such segments are considered an exact match during automatic substitution and the additional blanks at the end of the source segment are removed.

Use joined segments when found

Select this option to join segments. This can be useful if you want to analyze a document that you have worked with before and where you joined neighboring text segments before you translated them. During a new analysis run these segments are treated as separate units again. However, in the associated Translation Memory there is a match (translation) for the joined segments only. To find this match more easily during automatic substitution, this option joins neighboring segments in the document to be analyzed. Note that two neighboring segments are only joined if for the first segment in the sequence a fuzzy match has been found.

Ignore path in document names

Select this option if any path information in the document name is to be ignored when a document is checked for exact-exact matches.

No substitution if source and target segment is identical

Select this option to suppress the automatic substitution of segments when the source segment is identical with the target segment found in the **Translation Memory**.

Ignore memory proposals with comments

Select this option to suppress the automatic substitution of segments when the memory proposal has a comment.

Use TMO replacement list for IDDOC to DITA converted projects

Select this option to specify a TMO replacement list to be applied on IDDOC proposals for DITA

documents. Once this option is selected, the name of the replacement list can be typed in the list name field or be selected using the **Select....** button

List

The fully qualified file name of a TMO replacement list.

When you select both **Use exact context match...** and **Use latest exact match...**, the system combines both requirements and chooses the latest of the exact context matches. You then get fewer automatic substitutions than with **Use latest exact match...** but more than with **Use exact context match....**

Click **Analyze** to begin with the analysis of the document. As it can take some time to analyze the document, a window is displayed showing the progress of the operation.

Results

The document is segmented.

Depending on the options you selected, the document may already be partially translated, the **Translation Memory** updated, or a file created containing untranslated segments.

You can begin translating the document.

Editing analysis Profiles

analyzing **Editing analysis Profiles Translation Memory databases** An analysis profile allows the specification of analysis settings for a specific group of markup tables. When an analysis profile is specified in the analysis window the settings from the profile override the settings specified in the analysis window. When the markup table of the analyzed document does not match the markup tables defined in the profile the analysis profile settings are ignored and the settings from the analysis window are used instead.

Calling sequence

Select:

1. Activate the analysis window
2. Select the option
3. Select an analysis profile from the analysis profile list or enter the name of a new analysis profile
4. press the button

Results

The analysis profile window is displayed The "Analyze profile" window (see [Figure 45](#)) is displayed.

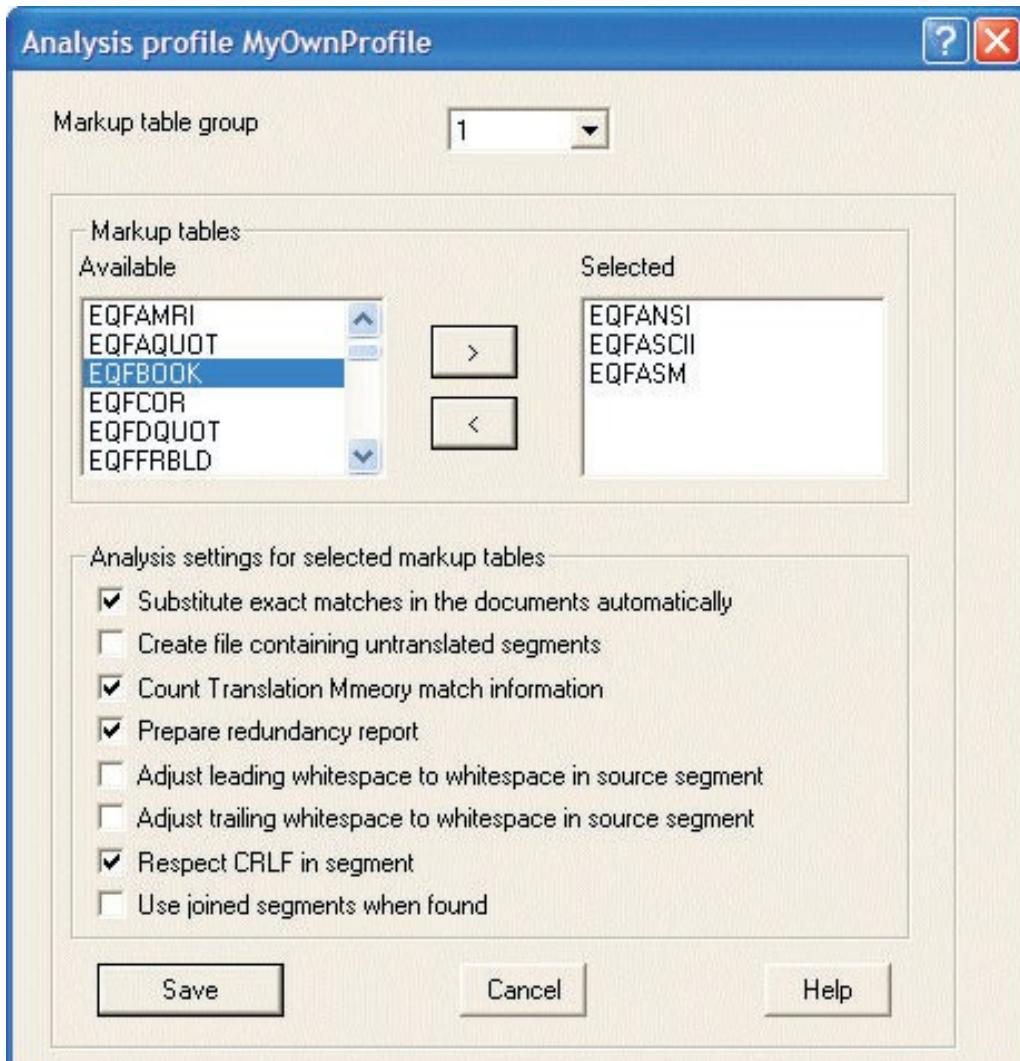


Figure 45. Analysis Profile window
[\[PIC\]Analyze Documents window](#)

Options and parameters

Markup table group

Select one of the markup table groups. The markup tables and the analysis settings for the selected group are displayed.

Available

This list shows the available markup tables. Markup tables already selected (in the same or another markup table group) are not listed here. To move a markup table to the selected list either double-click the markup table name or select the markup table name and press the ">" button.

Substitute exact matches in the documents automatically

Select this option to automatically replace original segments with the translations found in the Translation Memory associated with the document.

Create file containing untranslated segments

Select this option if you want OpenTM2 to create a separate file containing all untranslated segments. If you also selected **Substitute exact matches in the documents automatically**, this file contains only the remaining untranslated segments.

Count Translation Memory match information

Select this option to enable OpenTM2 to count the number and type of matches that are found in the Translation Memory for the document. The document is then segmented and the matches are counted but no segments are replaced with matches that are found in the Translation Memory.

Adjust leading white space to white space in source segment

Select this option if the text is to have the same leading white space (blank, tab stop, and line feed) before and after the automatic substitution.

Adjust trailing white space to white space in source segment

Select this option if the text is to have the same trailing white space (blank, tab stop, and line feed) before and after the automatic substitution.

Respect CRLF in segment

The Translation Memory may contain two matching target segments for a source segment, one with carriage return line feed (CRLF), and one without. If you select this option, the target segment containing CRLF is taken during automatic substitution.

Use joined segments when found

Select this option to join segments. This can be useful if you want to analyze a document that you have worked with before and where you joined neighboring text segments before you translated them. During a new analysis run these segments are treated as separate units again. However, in the associated Translation Memory there is a match (translation) for the joined segments only. To find this match more easily during automatic substitution, this option joins neighboring segments in the document to be analyzed. Note that two neighboring segments are only joined if for the first segment in the sequence a fuzzy match has been found.

- Click **Save** to save the analysis profile.
- Click **Cancel** to leave this window without saving the analysis profile.

Results

If you clicked on **Save**, the analysis profile is saved or created (if it is a new one) and can be used for the analysis process.

Usage of analysis profile

An analysis profile can be used when the analysis settings for documents must be different depending on the markup table of the document.

Normally you would select all documents of one markup table and perform the analysis for these documents and when select the documents which require different analysis settings and repeat the analysis for these documents.

With the help of an analysis profile this can be simplified. Just select one of the markup tables in markup table group 1 and select the settings required for this markup table, go to markup table group 2, select the second markup table and select the settings required for this markup table and save the profile. Select the profile in the analysis window and run the analysis process. For all documents with the markup table selected in the first markup table group the settings from the first markup table group are used and for all documents with a the markup table selected in the second group the settings from the second markup table group are used. For all other documents the settings from the analysis window are used.

Analyzing documents using dictionaries

analyzingdocuments using dictionaries terminology listscreating If you have *dictionaries* available containing vocabulary associated with the document, you can ask **OpenTM2** to generate two types of *terminology list*:

- *New terms* list.

terminology listslist of new terms

This list contains all terms in your document that are not defined in the referenced dictionaries. You can use this list of terms to add entries to these dictionaries.

- *Found terms* list.

terminology listslist of found terms

This list contains all terms in your document that are already defined in the referenced dictionaries. You can use this list of terms to fill a separate dictionary related to the document.

The generated terminology lists may contain terms that are unsuitable as terminology for your translation. **terminology listsexclusion lists** In this case, you can create the lists again using *exclusion lists* or *exclusion dictionaries* to get fewer terms in the lists.

If you are only interested in terms that occur more than once in a document, you can specify a minimum number of occurrences greater than 1.

To perform the creation of terminology lists, you specify your options and parameters in two subsequent windows.

Prerequisites

- Dictionaries and, optionally, exclusion lists must be available
- The folder containing the document must be opened

Calling sequence

Select:

1. One or more documents from the "Document List" window.
2. **Analyze...** from the **File** menu. The "Analyze Documents" window is displayed.

3. **Create Lists** The **List Settings** tab is created.
4. **List Settings tab**

The "List Settings" page (see [Figure 46](#)) is displayed.

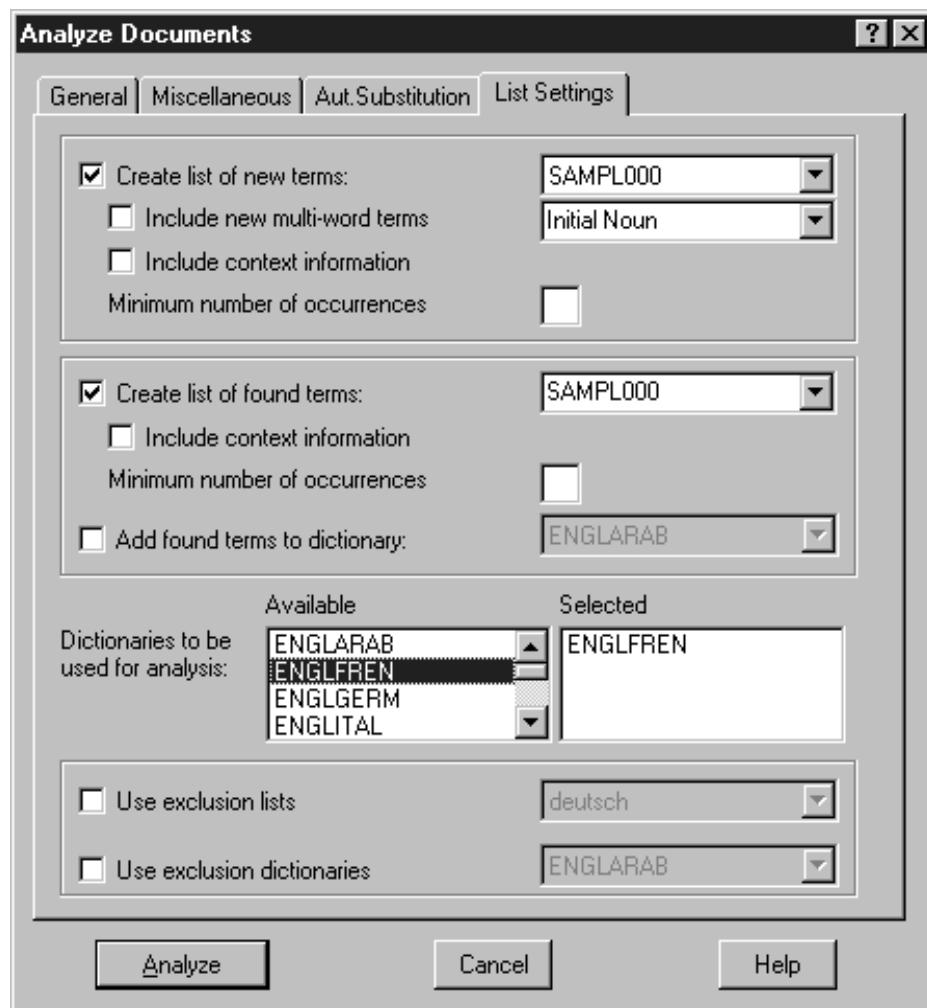


Figure 46. Analyze Documents window — List Settings page
[\[PIC\]](#)Figure shows Analyze Documents window — List Settings page

Options and parameters

To create a new terms list, select::

Create list of new terms

creatinglists of new terms Select this option to create a list of new terms. The system provides the folder name as the default name for the list to be generated. You can specify any other name. For this type of list, you can also select:

Include new multiword terms

Select this option to include multiword terms that are not hyphenated, such as data management, in the list of new terms. Choose whether the first word of these terms can be a noun, an adjective, or either of them.

Include context information

If a new term is found, the system can save the original segment (containing the term) as *context* information. This option is useful if you intend to copy the new terms to a dictionary that can contain context information.

Minimum number of occurrences

Specify how often a term must occur in the document so that it is included in the new terms list.

To create the list of all terms of a document that are also in selected dictionaries, select:

Create list of found terms

creatinglists of found terms Select this option to create a list of found terms. The system provides the folder name as the default name for the list to be generated. You can specify any other name. For this type of list, you can also select:

Include context information

If a term is found in one of the selected dictionaries, the system can save the original segment (containing the term) as *context* information. This option is useful if you intend to copy the found terms to a dictionary that can contain context information.

Minimum number of occurrences

Specify how often a term must occur in the document so that it is included in the found terms list.

Add found terms to dictionary

To copy the found terms to a dictionary, select a dictionary from this list box.

For both new terms and found terms lists, you must specify:

Dictionaries to be used for analysis

Select the dictionaries to be used for generating terminology lists from the **Available** list box. The dictionaries are listed in the **Selected** list box in the order in which you select them, and the dictionaries are searched in this order. You must select at least one dictionary, and you can select up to 10.

If needed, you can limit the generation of terminology lists by the following options:

Use exclusion lists

If you have terms you want to exclude from the lists to be generated, use an exclusion list. For each language for which you installed the language support, OpenTM2 already provides an exclusion list. It contains so-called noise terms. Select the exclusion lists to be used from the list box.

Use exclusion dictionaries

If you have a dictionary that contains well-defined terms that you want to exclude from the terminology lists to be generated, select it from the list box.

Click **Set** to return to the "Analyze Documents" window.

To begin analysis, click **Analyze**.

Results

The document is segmented.

Depending on the options you selected, new terms lists and found terms lists are created, and can be modified and used for dictionary updates. For information on how to process new terms lists and found terms lists, see [Working with terminology lists](#).

Analyzing documents using machine translation (MT)

You can begin translating the document.

Maintaining abbreviation lists

abbreviation listmaintaining OpenTM2 uses language-specific abbreviation lists to identify correct endings of a sentence. This prevents that the period after an abbreviation is interpreted as the end of a sentence during analysis, for example in a sentence like For details see books, films, magazines, etc. and keep the subject in mind.. The period after etc. is not considered to be the end of the sentence. OpenTM2 provides default abbreviation lists for each source language supported. You can add new abbreviations, or delete or modify existing ones.

Prerequisites

The "Language List" window must be active (**Display Language List** from the Utilities menu).

- The language you want to work with must be in active state.

Calling sequence

Select:

1. An active language
2. **Open** from the **File** menu or double-click the name of the language

The list of abbreviations for the selected language is displayed. For example, **Edit Abbreviations for: German(national)** if you selected this language. You can now make your changes.

Options and parameters

Abbreviations list

To update an abbreviation, overtype it. To add a new one, switch to Insert mode and press Enter at the line before which you want to insert it. To delete an abbreviation, use the Backspace key. Note that you end the abbreviation with a full stop.

Save

To save any changes and close this window.

Cancel

To close this window without saving any changes.

Help

To get information on how to interact with this window.

Results

The language-specific list of abbreviations is updated and will be used for subsequent analysis runs. Depending on the language you selected, abbreviations have been changed, added, or removed from the abbreviation list for this language.

Closing a document

documentclosing stop translating ending translations saving translations leaving the **Translation Environment** returning to the main window To translate a document, you must open it so that the **Translation Environment** for this document is displayed.

Closing a document means to stop translating a document and to leave the **Translation Environment**.

Calling sequence

Select **Close** from the **Translation Environment** system menu.

This takes you back to the **OpenTM2** main window.

Results

The translation document, the **Translation Memory**, and the dictionaries used are saved in their current status. You can start any other **OpenTM2** task. The translation of this document can be continued at any time.

Counting the number of words in a document

countingwords documentcounting words in documentcounting markup tags in countingmarkup tags in a document **OpenTM2** provides a facility for counting the number of words and markup tags in a document at any time. The result is shown in a window and can be saved in a file.

To count the number of words in all the documents of a folder, see [Counting the words in all documents of a folder](#).

Prerequisites

At least one document must exist in the selected folder.

Calling sequence

Select:

1. At least one document from the "Document List" window
2. **Count Words...** from the **Utilities** menu

The "Count Words" window (see [Figure 47](#)) is displayed.

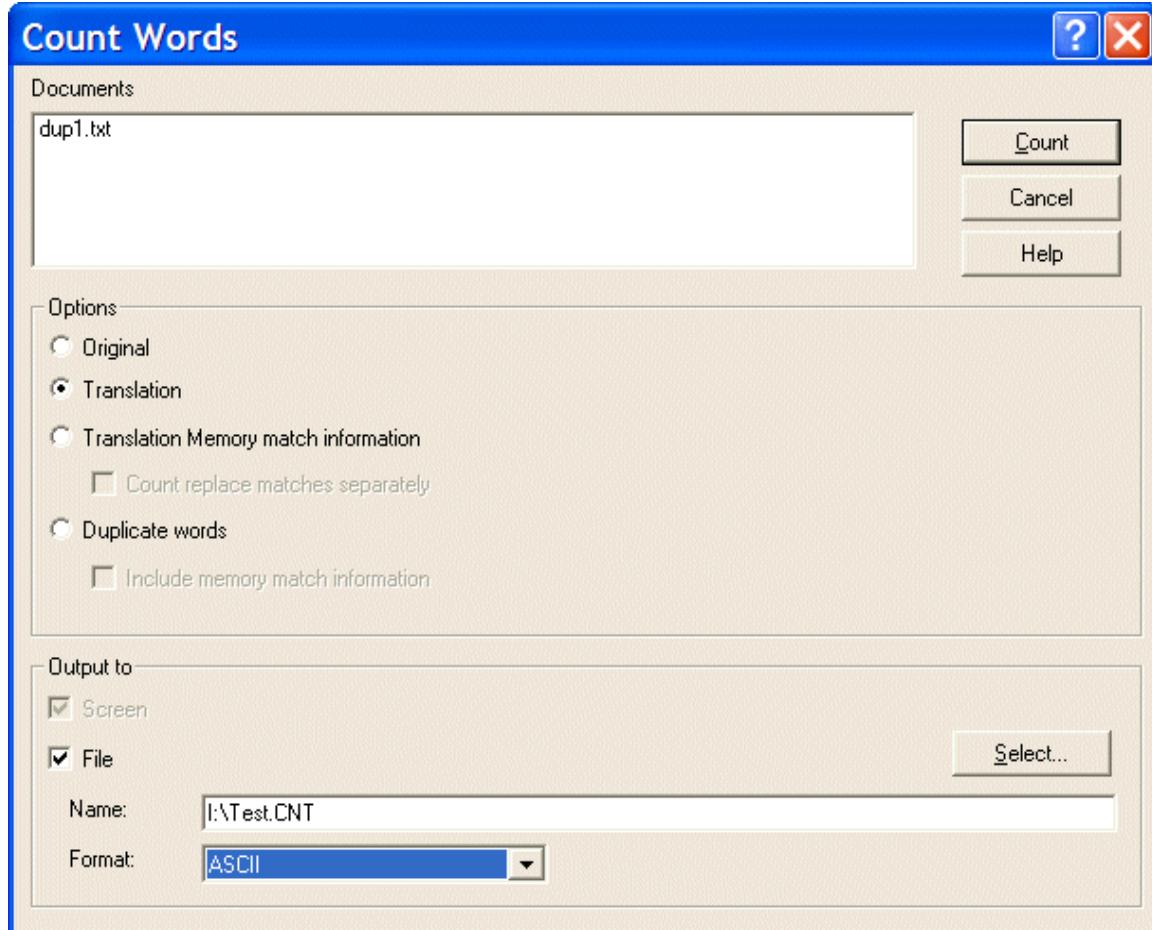


Figure 47. Count Words window

Options and parameters

Documents

This box contains the list of the documents you selected in the "Document List" window.

Options

One of the following options can be selected at a time. Select:

Original

To count the number of words in the document before translation.

Translation

To count the number of words in the translated document.

Translation memory match information

See [Counting the number and type of matches in a document](#) for an explanation.

Duplicate words

See [Counting duplicate words in a document](#) for an explanation.

Fuzzy matches by class

See for an explanation.

Output to

In this block you define where you want the system to save the results:

Screen

The result of the word count is displayed on your screen. This option is selected automatically.

File

Select this option to also store the count results in a file. You can:

- Accept the proposed file name.
- Type a file name of your choice.
- Click **Select...** to display the "Select File" window from where you can choose the file name without having to type it (for a detailed description, see [...\).](#)

- Select the format of the count result file. The supported formats are
 - ASCII (plain text)
 - HTML
 - XML
 -

Click **Count** to begin counting words.

Results

If you selected **Original** and output to **Screen**, the result is displayed in the "Word Count Results - Original" window (see) with the following columns for each original document:

- **Documents**
The name of the document
- **Untranslated**
The number of words
- **Markup**
The number of markup tags
- **Total**
The number of words and markup tags

If you selected **Translation** and output to **Screen**, the result is displayed in the "Word Count Results - Translation" window (see [Figure 2](#)) with the following columns for each translated document:

- **Documents**
The name of the document.
- **Translated**
The number of original words already translated. Note that copying a marked part of a **Translation Memory** proposal (see [Moving around marked text](#)) is counted as one character.
- **Untranslated**
The number of original words not yet translated.
- **Target words**
The number of translated words in the translated document.
- **Markup**
The number of markup tags.
- **Total**
The number of translated words and markup tags.

Documents	Translated	Untranslated	Target Words	Markup
dup1.txt	7	14	6	0
=====..	=====	=====	=====	=====
Total	7	14	6	0

Figure 48. Word Count Results - Translation window

If you selected the **File** output option, the results are stored in the same format in the specified file.

To return to the previous window, close the respective "Word Count Result" window.

Counting duplicate words in a document

countingwords documentcounting duplicate words in OpenTM2 provides a facility for counting duplicate words for one or more documents. The result is shown in a window and can be saved in a file. Duplicate words are words in segments which are contained more than once in the documents and will become exact matches once the first occurrence of such a segment has been translated.

To count the number of words in all the documents of a folder, see [Counting the words in all](#)

[documents of a folder.](#)

Using the "**Include memory match information**" option, information on memory matches can be included in the created results.

Prerequisites

At least one document must exist in the selected folder.

Calling sequence

Select:

1. At least one document from the "Document List" window
2. **Count Words...** from the **Utilities** menu

The "Count Words" window (see [Figure 49](#)) is displayed.

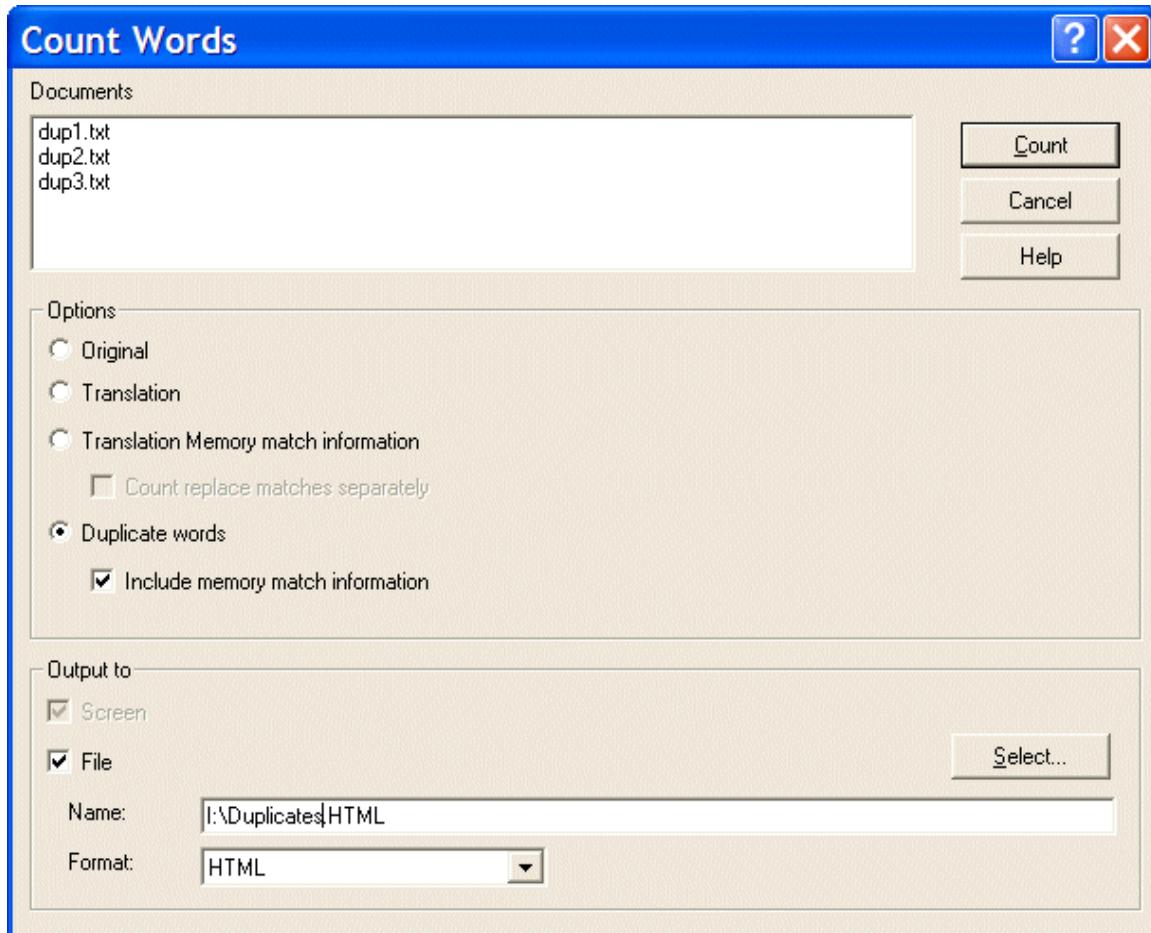


Figure 49. Count Words window

Options and parameters

Documents

This box contains the list of the documents you selected in the "Document List" window.

Options

Select:

- **Duplicate words**

To count the number of duplicate words in the selected documents.

- **Include memory match information**

To show information on memory matches in the results.

Output to

In this block you define where you want the system to save the results:

Screen

The result of the word count is displayed on your screen. This option is selected automatically.

File

Select this option to also store the count results in a file. You can:

- Accept the proposed file name.

- Type a file name of your choice.
- Click **Select...** to display the "Select File" window from where you can choose the file name without having to type it (for a detailed description, see [...\).](#)
- Select the format of the count result file. The supported formats are
 - ASCII (plain text)
 - HTML
 - XML

Click **Count** to begin counting words.

Results

Result without "Include memory match information" option selected. The results of the word count are shown, for example, in the "Word Count Results - Duplicate Words" window (see [Figure 2](#)).

Documents	Translated	Untranslated	Duplicate Words	To be translated	Markup
dup1.txt	7	14	0	14	0
dup2.txt	0	20	15	5	0
dup3.txt	0	23	10	13	0
=====	=====	=====	=====	=====	=====
Total	7	57	25	32	0

Figure 50. Word Count Results - Duplications Words window
[\[PIC\]Figure shows Word Count Results - Duplicate Words window](#)

The window contains the following columns:

- **Documents**
The name of the document.
- **Translated**
The number of translated words in the documents.
- **Untranslated**
The number of untranslated words in the documents.
- **Duplicate words**
The number of words in duplicate segments. The first occurrence of these segments is counted in the "To be translated" column.
- **To be translated**
The number of words which have to be translated manually.
- **Markup**
The number of tags in the documents.

If you selected the **File** output option, the results are stored in the same format in the specified file.

To return to the previous window, close the respective "Word Count Result — Duplicate Words" window.

Results

Result with "Include memory match information" option selected. The results of the word count are shown, for example, in the "Word Count Results - Duplicate Words" window (see [Figure 3](#)).

Documents	Duplicate Words (Repetitions)	95-99% Fuzzy	75-94% Fuzzy	0-74% Fuzzy	Total Untranslated	Exact matches (100% Matches)
dup1.txt	0	0	14	7	21	0
dup2.txt	15	0	0	5	20	0
dup3.txt	10	0	5	8	23	0
=====	=====	=====	=====	=====	=====	=====
Total	25	0	19	20	64	0

Figure 51. Word Count Results with memory match information- Duplications Words window
[\[PIC\]Figure shows Word Count Results with memory match information - Duplicate Words window](#)

The window contains the following columns:

- **Documents**

The name of the document.

- **Duplicate words (Repetitions)**

All words in segments, which had already one occurrence in one of the document(s). The first occurrence of the segment is listed in one of the fuzzy columns and any additional occurrence of the segment is listed in the duplicate words column.

- **95-99% Fuzzy**

The number of words in segments for which there is a fuzzy match in the memory with a fuzziness of 95% up to 99% or segments for which another segment would become a 95-99% fuzzy match proposal during the translation of the document(s).

- **75-94% Fuzzy**

The number of words in segments for which there is a fuzzy match in the memory with a fuzziness of 75% up to 94% or segments for which another segment would become a 75-94% fuzzy match proposal during the translation of the document(s).

- **0-74% Fuzzy**

The number of words in segments for which there is a fuzzy match in the memory with a fuzziness of 0% up to 74% or segments for which another segment would become a 0-74% fuzzy match proposal during the translation of the document(s).

- **Total untranslated**

The total number of words in segments which have no exact match in the Translation Memory. This column contains the total of the duplicitae words column and the fuzzy columns.

- **Exact matches (100% Matches)**

The number of words in segments for which there is one or more exact match in the Translation Memory.

If you selected the **File** output option, the results are stored in the same format in the specified file. To return to the previous window, close the respective "Word Count Result — Duplicate Words" window.

Counting the number and type of matches in a document

countingmatches documentcounting matches in Translation Memorycounting matches from Translation Memory matchescounting matchescounting OpenTM2 provides a facility for counting the number and type of **Translation Memory** matches for one or more documents. The result is shown in a window and can be saved in a file.

Prerequisites

At least one document must exist in the selected folder.

Calling sequence

Select:

1. At least one document from the "Document List" window
2. **Count words...** from the **Utilities** menu

The "Count Words" window (see [Figure 52](#)) is displayed

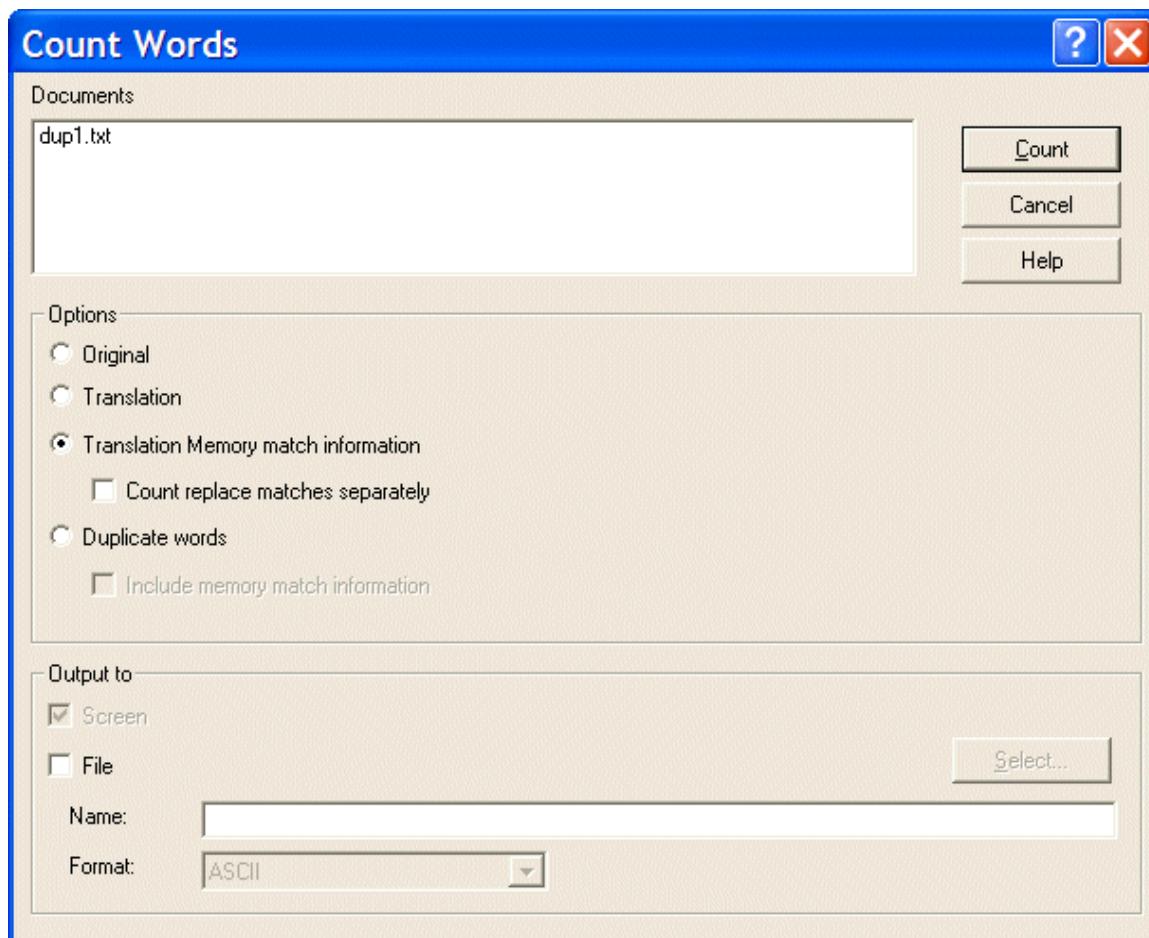


Figure 52. Count Words window

Options and parameters

Documents

This box contains the list of the documents you selected in the "Document List" window.

Options

Select:

- **Translation Memory match information**
To count the number and type of matches found in the **Translation Memory** for the selected documents.
- **Count replace matches separately**
To show the matches replaced automatically, in separate columns.

Output to

In this block you define where you want the system to save the results:

Screen

The result of the word count is displayed on your screen. This option is selected automatically.

File

Select this option to also store the count results in a file. You can:

- Accept the proposed file name.
- Type a file name of your choice.
- Click **...** to display the "Select File" window from where you can choose the file name without having to type it (for a detailed description, see **...**).
- Select the format of the count result file. The supported formats are
 - ASCII (plain text)
 - HTML
 - XML

Click **Count** to begin counting the matches.

Results

The results of the word count are shown, for example, in the "Word Count Results - Translation Memory Matches" window (see [Figure 53](#)).

Word Count Results - Translation Memory Matches							
Documents	Total	Exact-Exact	Exact (1)	Exact (2+)	Fuzzy	MachMatch	No match
D dup1.txt	21	7	0	0	14	0	0
=====	=====	=====	=====	=====	=====	=====	=====
D Total	21	7	0	0	14	0	0

Figure 53. Word Count Results - Translation Memory window

The window contains the following columns:

- **Documents**
The name of the documents.
- **Total**
The number of words in the documents.
- **Exact-Exact**
The number of words in segments with identical matches.
- **Exact (1)**
The number of words in segments having exactly one proposal.
- **Exact (2+)**
The number of words in segments having more than one exact proposal.
- **Fuzzy**
The number of words in segments with fuzzy proposals. If you did not select **Count replace matches separately**, the number displayed also includes the number of words in segments automatically replaced and automatically replaced in a fuzzy match.
- **MachMatch**
The number of segments translated by machine.
- **No match**
The number of words in segments with no proposal.

If you selected **Count replace matches separately**, you get the following fields in addition:

- **Replace**
The number of words in segments automatically replaced.
- **FuzzyRepl**
The number of words in segments automatically replaced in a fuzzy match.

If you selected the **File** output option, the results are stored in the same format in the specified file.

Note that the criteria for fuzzy matches can be customized, as described in [#unresolvedid/viewsysprop](#). This also influences the "No match" counter.

fuzzy match overlap ratio impact on counting matches If there is no **Translation Memory** match data available for a document, for example because the document has not been analyzed yet or neither the **Substitute exact matches in the documents automatically** options nor the **Count Translation Memory match information** option was selected during analysis, the "Word Count Results - Translation Memory Matches" window contains the text No match

To return to the previous window, close the "Word Count Result — Translation Memory Matches" window.

Counting words in segments with fuzzy matches

countingmatches document Counting words in segments with fuzzy matches OpenTM2 provides a facility for counting words in segments with fuzzy matches. The result is shown in a window and can be saved in a file. The words in segments with fuzzy matches are grouped in classes. The column Class-0 lists the number of words in segments with fuzzy matches which are only different in white space or inline tagging. The column Class-1 lists the number of words in segments with one different word , class-2 with two different words and so on. To count the number of words in all the documents of a folder, see [Counting the number of words in a document](#)

Prerequisites

At least one document must exist in the selected folder and the document has to be analyzed.

Calling sequence

Select:

1. At least one document from the "Document List" window
2. from the menu

The "Count Words" window (see [Figure 54](#)) is displayed.

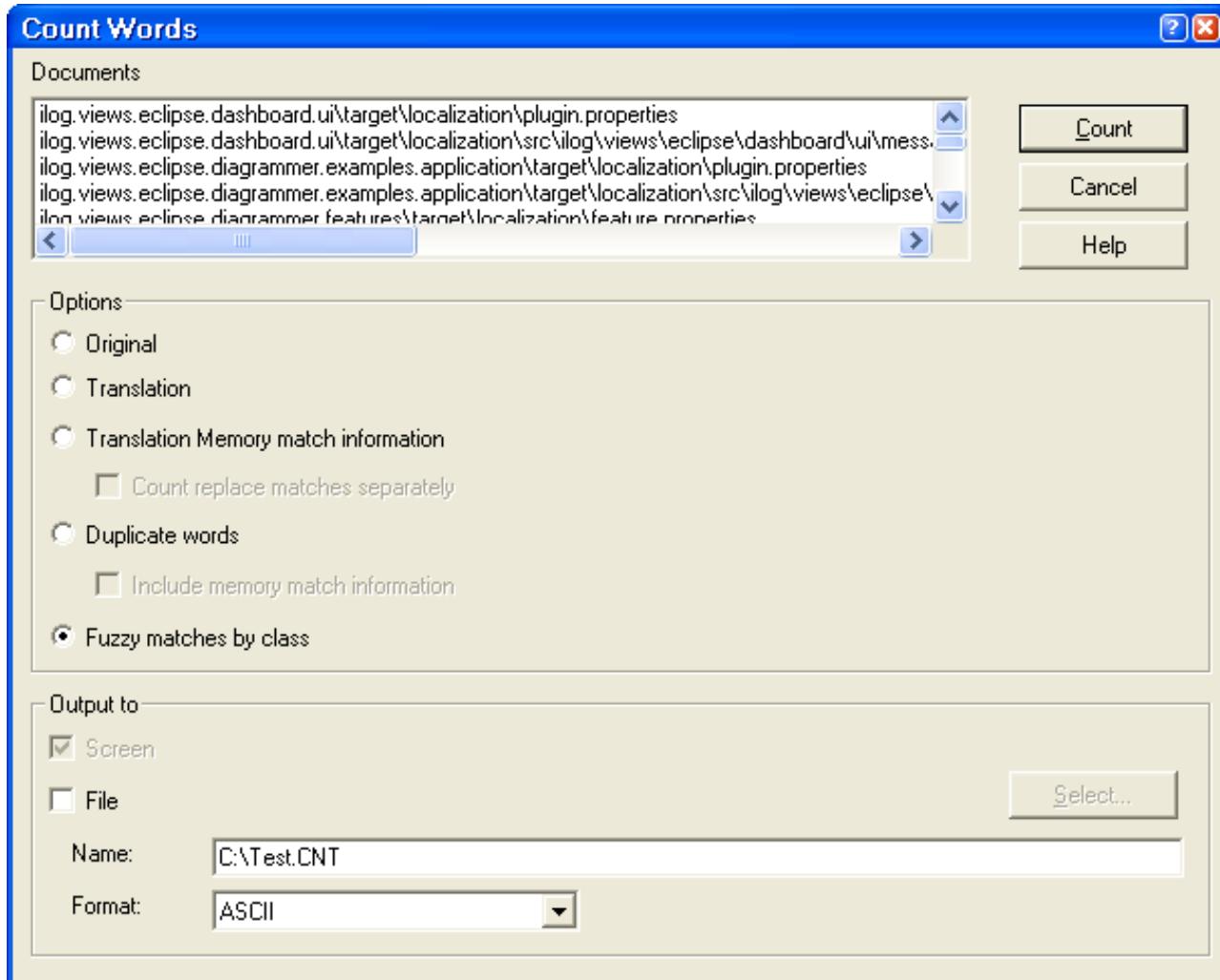


Figure 54. Count Words window

[PIC]Figure shows Count Words window

Options and parameters

Documents

This box contains the list of the documents you selected in the "Document List" window.

Options

Select:

- **Fuzzy matches by class**
To count the number of words in segments with fuzzy matches in the selected documents.
- **Output to**
In this block you define where you want the system to save the results:
- **Screen**
The result of the word count is displayed on your screen. This option is selected automatically.
- **File**
Select this option to also store the count results in a file. You can:

Output to

In this block you define where you want the system to save the results:

Screen

The result of the word count is displayed on your screen. This option is selected automatically.

File

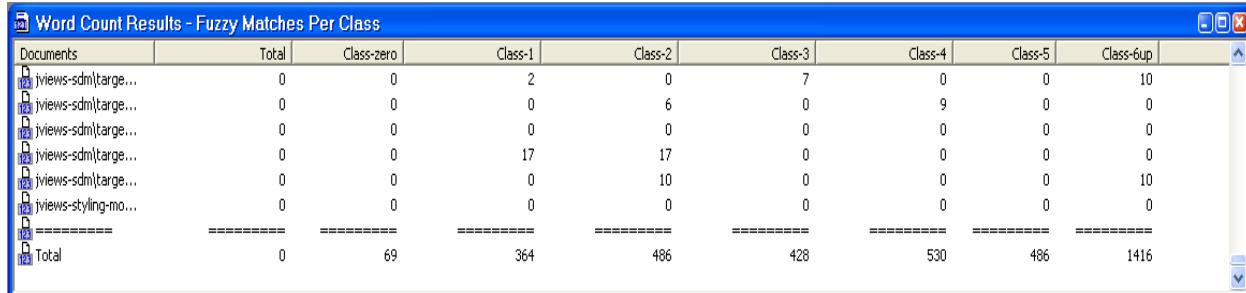
Select this option to also store the count results in a file. You can:

- Accept the proposed file name.
- Type a file name of your choice.
- Click  to display the "Select File" window from where you can choose the file name without having to type it (for a detailed description, see ).
- Select the format of the count result file. The supported formats are
 - ASCII (plain text)
 - HTML
 - XML

Click **Count** to begin counting words

Results

The results of the word count are shown, for example, in the "Word Count Results – Fuzzy Matches per Class" window (see [Figure 55](#)).



Documents	Total	Class-zero	Class-1	Class-2	Class-3	Class-4	Class-5	Class-6up
jviews-sdm target...	0	0	2	0	7	0	0	10
jviews-sdm target...	0	0	0	6	0	9	0	0
jviews-sdm target...	0	0	0	0	0	0	0	0
jviews-sdm target...	0	0	17	17	0	0	0	0
jviews-sdm target...	0	0	0	10	0	0	0	10
jviews-styling-mo...	0	0	0	0	0	0	0	0
=====	=====	=====	=====	=====	=====	=====	=====	=====
Total	0	69	364	486	428	530	486	1416

Figure 55. Word Count Results – Fuzzy Matches per Class window

[PIC]Figure shows Word Count Results – Fuzzy Matches per Class window

The window contains the following columns:

- **Documents**
The name of the documents.
- **Total**
The number of words in the documents.
- **Class-zero**
The number of words in segments with fuzzy matches where no source words are different.
- **Class-1**
The number of words in segments with fuzzy matches where one source word is different.
- **Class-2**
The number of words in segments with fuzzy matches where two source words are different.
- **Class-3**
The number of words in segments with fuzzy matches where three source words are different.
- **Class-4**
The number of words in segments with fuzzy matches where four source words are different.
- **Class-5**
The number of words in segments with fuzzy matches where five source words are different.
- **Class-6up**
The number of words in segments with fuzzy matches where six or more source words are different.

If you selected the output option, the results are stored in the same format in the specified file.

To return to the previous window, close the respective "Word Count Result – Fuzzy Matches per Class" window.

Dealing with specific document formats

OpenTM2 can work with documents that originate from a variety of word processors and formats.

Most word processors include facilities for converting document files from one document format to

another. If the file format of your word processor is not explicitly supported by **OpenTM2**, you may be able to use these conversion facilities to convert your file to a format that is supported directly by **OpenTM2**.

Sometimes, it can be helpful to have a printed version of the original document available to get a better picture of the whole document.

It is recommended that you check your document after translation for any changes concerning the layout. You can edit and print it with the word processor that was used for creating the original document. Minor format corrections might be necessary because translated text has not always the same length as the corresponding original text.

Ami Pro documents

OpenTM2 directly supports the document format of Ami Pro Version 2 or 3, which usually has the file extension SAM. Import such documents using the EQFAMI markup table. If they were designed for processing by Lotus^(R) Notes^(TM), import them using the LOTUSAMI markup table.

Do not change, delete, or insert markup data outside translatable segments. Inline tags should be handled carefully.

The start of highlighting like bold, italics, underline, or similar is indicated with <+, any character, and >. The end of this markup data is indicated with <-, the same character as at the start of this markup tag, and >.

Tables, headers, and footers appear at the beginning of the document for translation.

ANSI documents

OpenTM2 provides the following markup tables to import different ANSI (Windows ISO 8859-x) documents:

- EQFALINE for files in which everything is translatable.
- EQFAQOUT for files in which translatable text is enclosed in single quotes.
- EQFAMRI for files in which translatable text is enclosed in double quotes.
- EQFANSI for ANSI documents. This markup table is combined with a user exit.

The EQFALINE markup table is used for ANSI documents that are completely translatable. In such documents, each line is translated separately. This means that text segmentation is based on a line basis. Such documents do not contain comments.

The EQFAQOUT markup table is used for files in which translatable text is enclosed in single quotes ('') and the EQFAMRI markup table for files in which translatable text is enclosed in double quotes (""). Comments, which are not translatable, begin with two backslashes (\\\) and end with the end of the current line or begin with * and end with *\|. In the latter case, comments can span several lines. Replacement variables that start with a percent sign (%) and have a well-defined format like %8.8s or %ld, are not translatable.

ASCII documents

OpenTM2 provides the following markup tables to import different ASCII documents, which are text files without any formatting tags:

- EQFLINE for files in which everything is translatable
- EQQUOTE for files in which translatable text is enclosed in single quotes
- EQFMRI for files in which translatable text is enclosed in double quotes
- EQFASCII for ASCII documents

The EQFLINE markup table is used for ASCII documents that are completely translatable. In such documents, each line is translated separately. This means that text segmentation is based on a line basis. Such documents do not contain comments.

The EQQUOTE markup table is used for files in which translatable text is enclosed in single quotes ('') and the EQFMRI markup table for files in which translatable text is enclosed in double quotes (""). Comments that are not translatable begin with two backslashes (\\\) and end with the end of the current line or begin with * and end with *\|. In the latter case, comments can span several lines. Replacement variables that start with a percent sign (%) and have a well-defined format like %8.8s or %ld, are not translatable.

Assembler documents

OpenTM2 allows you to translate text contained in Assembler files, that is, information that is enclosed by single quotes. For example:

```
*2.....10.....16.....72.....  
DC C'THIS IS A ONE LINE TEXT' DC C'This is  
a two line text string that will be presented x under  
OpenTM2 on one line'
```

Import these documents into a **OpenTM2** folder, using the **EQFASM** markup table.

Text spread over more than one line is shown on one line in the "Translation" window. Translated text that does not fit into one source record is spread into several lines. A continuation character is placed in column 72, and the continuation begins in column 16.

BookMaster (R) and Information Presentation Facility documents

Import these documents using the **EQFBOOK** markup table.

FrameBuilder files

OpenTM2 supports FrameBuilder Version 5.5 files, which usually have the file extension MIF.
Import these files using the **EQFFRBLD** markup table.

HTML documents

Import HTML level 2.0 and 3.0 documents using the **EQFHTML2** markup table and level 4.0 documents using **EQFHTML4**. The latter markup table is combined with a user exit.

Interleaf files

OpenTM2 supports Interleaf (ASCII) files, which usually have a file extension of DOC. Import these files using the **EQFINT2** markup table.

Lotus (R) Notes (TM) documents

Import Lotus (R) Notes (TM) documents using the **LOTUSNGD** markup table.

Microsoft (R) PowerPoint (R) documents

Import Microsoft (R) PowerPoint (R) documents using the **EQFPPT** markup table.

Microsoft (R) Word for Windows (R) documents

OpenTM2 supports the following document formats of Microsoft (R) Word for Windows (R):

- Word Version 2.0
- Word Version 6.0
- Word Version 7.0 (Word 97)
- Word Version 8.0 (Word 98)
- Word Version 9.0 (Word 2000)

Import these documents using the **EQFMSWRD** markup table.

Note:

The markup tables **EQFWORD** and **EQFWORD6** are no longer supported. Therefore, analyze existing Word documents again using the new markup table **EQFMSWRD**.

RTF documents

Rich Text Format (RTF) is a document format supported by many word processors. Import these documents using the **EQFRTF** markup table.

Select **Check for changes of inline tags** during translation to ensure that braces ({}) are not misplaced or deleted unintentionally. Otherwise the translated document may no longer be accepted by the original word processor.

SGML documents

Import SGML documents that were designed for processing by Lotus (R) Notes (TM) using the **EQFSGML** markup table.

Unicode documents

Unicode enables you to display, and work with, documents in **OpenTM2** that have different character sets, such as German and Japanese, or different code pages.

OpenTM2 provides the following markup tables to import Unicode documents:

- **EQFUTF8** for files which are encoded in UTF-8 format
- **UNICODE** for files which are encoded in UTF-16 format

To use Unicode you must import the documents using the **UNICODE** or the **EQFUTF8** markup table.

Windows^(R) help files

These files are used to define help texts for programs running under Windows^(R). They can be saved as RTF documents.

Import these documents using the EQFRTF markup table.

WordPerfect documents

OpenTM2 supports WordPerfect document formats that were created with WordPerfect Version 5.x, Version 6.0, and Version 6.1. You can import these documents into **OpenTM2** using the **EQFWP** (for WordPerfect Version 5.x documents) or **EQFWP6** (for WordPerfect Version 6.0 and 6.1 documents) markup table.

When you have translated the document, you can export it in external format and load it back into WordPerfect to generate a new index and new table of contents, for example, to print and review the translation.

Note:

During translation, segments spanning more than two lines in the "Translation" window are joined including a blank space. As an exception to this rule, a blank is not inserted if an inline tag (such as [bold] or [italic]) is found at the end of a line. However, if you need a blank between the tag and the first word of the following line, it is recommended to join the lines such that no line ends with a tag or to start the following line with a blank before the first word.

Deleting a document

deletingdocuments documentdeleting erasing translations When you delete a document, you also delete the translated version of the document. However, segments you have translated are preserved in the **Translation Memory**.

Prerequisites

The folder containing the document must be open.

Calling sequence

Select:

1. One or more documents from the "Document List" window of the appropriate folder. To delete all documents in a folder, choose **Select all** from the **File** menu.
2. **Delete** from the **File** menu.

You must confirm that you want to delete each selected document.

Results

The documents are deleted.

Exporting a document

exportingdocuments documentexporting returningtranslation to customer transferring translation to another user Exporting documents allows you to share documents with other users. Documents can be exported at any time: before, during, and after translation. Partially translated documents can then be translated further by the recipient.

To transfer a document to another **OpenTM2** user, it is recommended that you export it in **OpenTM2** format, so that the recipient receives the document in its original format. If the recipient needs to have the document in a specific word-processor format, export it in external format.

For proofreading or validation the translated document can be exported as validation document

In addition, you can export translated glossary terms contained in a document to a Lotus^(R) Notes^(TM) database.

You can start the export procedure from within **OpenTM2**, or you can export documents in external format by entering the EQFCMD command and its parameters in the command area of your operating system. You may want to do this, for example, if you have many large documents to export. This command and its parameters are explained in [Exporting documents](#).

Prerequisites

The "Document List" window of the respective folder is active.

Calling sequence

Select:

1. One or more documents from the "Document List" window
2. **Export** from the **File** menu

The "Export Documents" window is displayed (see [Figure 56](#)).

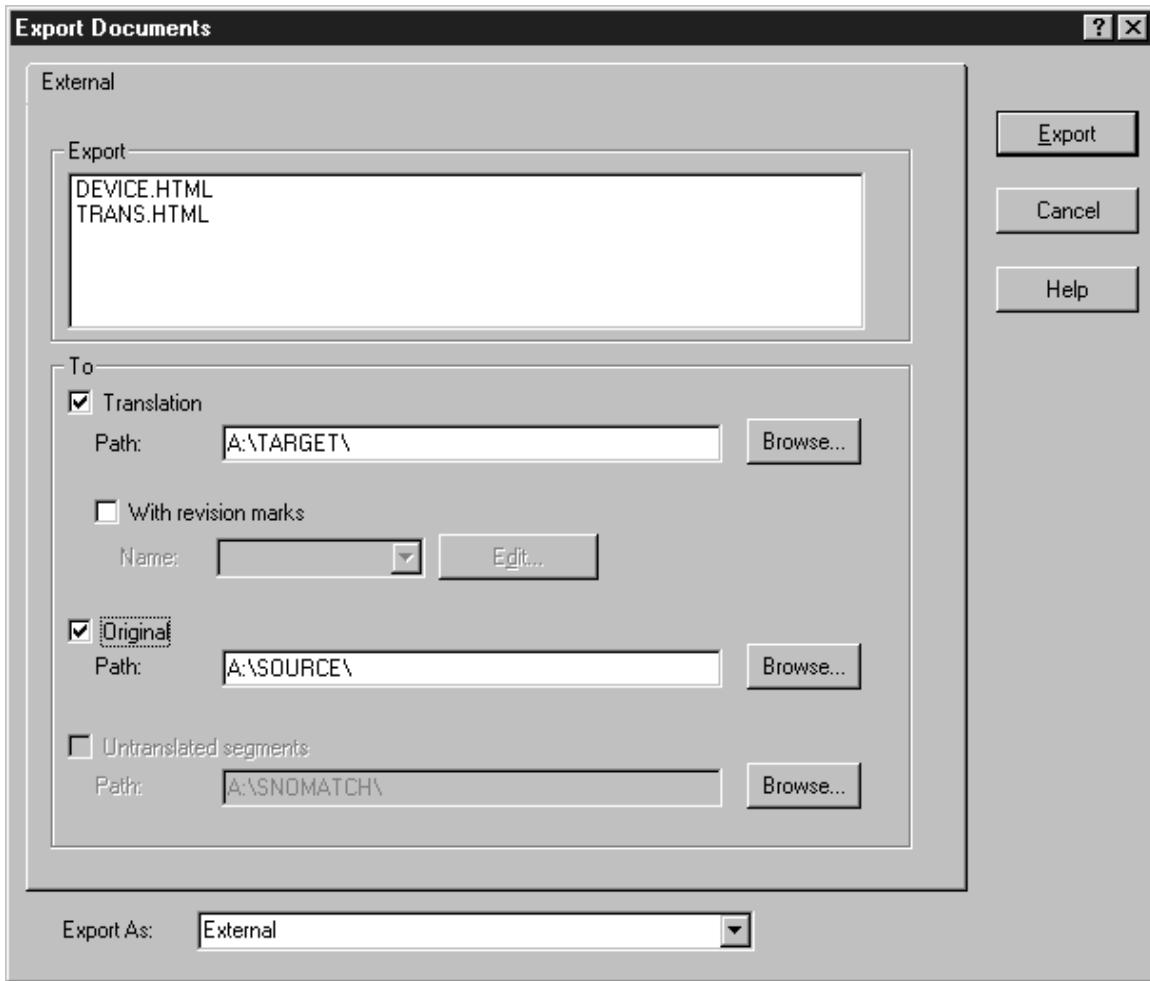


Figure 56. Export Documents window

Exporting a document in OpenTM2 format

Calling sequence

To export a document that is in **OpenTM2** format, select **OpenTM2 Format** from **Export as**. The "**OpenTM2 Format**" page is displayed (see [Figure 57](#)).

Figure 57. Export Documents window (OpenTM2 format)

Options and parameters

Export

This box contains a list of the documents you selected in the "Document List" window.

To

Specify the drive to which you want to export the documents. You can also click **Browse** to display the "Browse for Folder" window on which you can select the appropriate drive.

Click **Export** to start document export.

Results

The document is exported and stored in **OpenTM2** on the selected drive on your workstation in a separate subdirectory of the \eqf\export path under the file name you specified. For example, if the document device.html is exported from the folder samplehtml3, it is stored as \eqf\export\samplehtml3.f00\device.html. The document can be passed to other **OpenTM2** users for further processing.

Exporting a document in external format

exporting documents in external format document exporting in external format You use the external format to distribute a translated document for checking, for example.

Calling sequence

To export a document in external format, select **External Format** from **Export as**. The "External Format" page is displayed (see [Figure 58](#)).

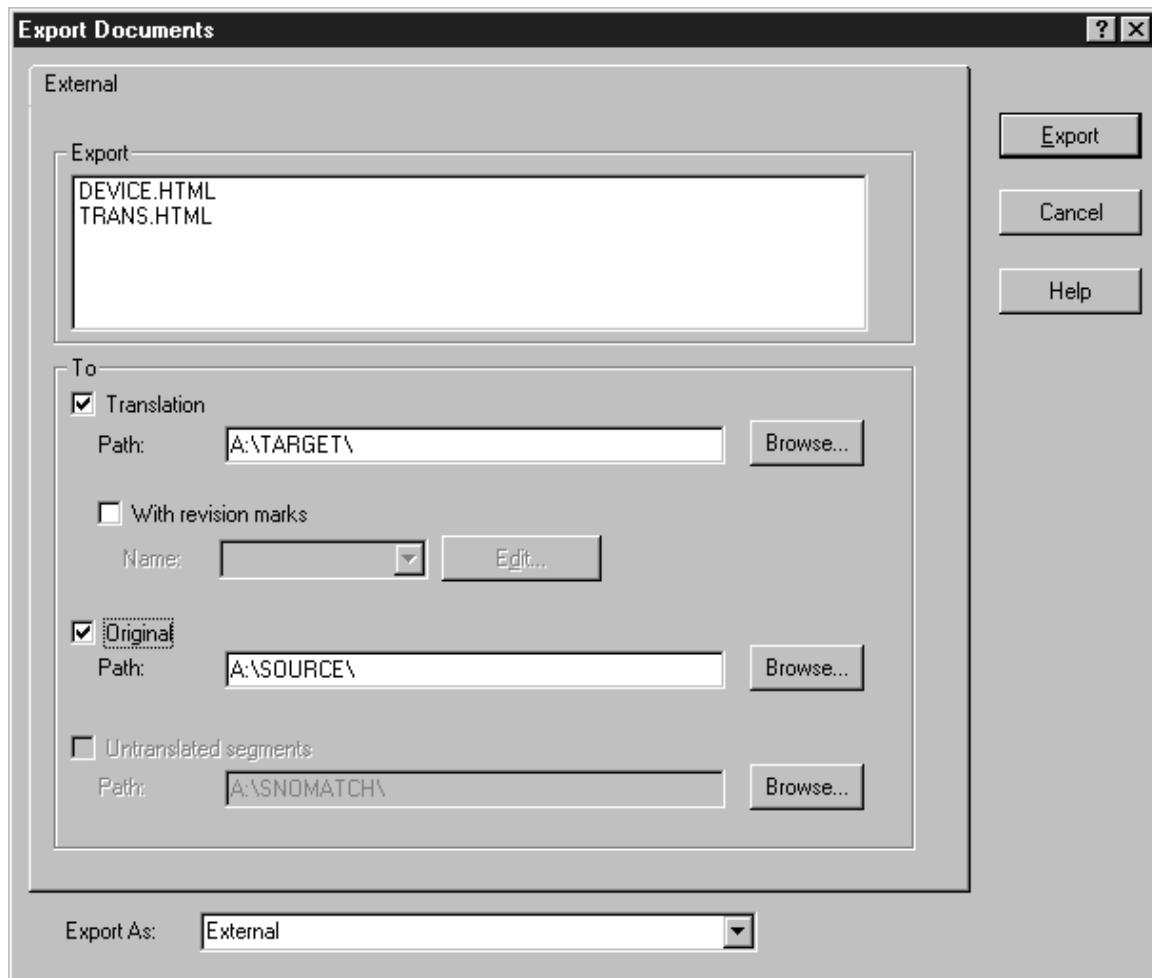


Figure 58. Export Documents window (External format)

Options and parameters

Export

This box contains a list of the documents you selected in the "Document List" window.

Translation

Select this option to export the documents in their translated version.

Path

Type the path name where you want the translated documents to be stored on the selected drive. Use a trailing backslash (\). You can also click **Browse** to display the "Browse for Folder" window on which you can select the appropriate path.

With revision marks

Select this option to have the parts of the exported document that have been newly translated marked. Revision marks can be individually defined by you. To specify a set of revision marks to be used, do one of the following:

- Type the name of an existing set of revision marks or a new name.
- Select an existing name from the list box.

To change an existing set of revision marks or to define new revision marks, click **Edit...**. This takes you to the "Edit Revision Marks" window where you define your revision marks (see [Editing document revision marks for export](#)).

Original

Select this option to export the documents in their original version.

Path

Type the path name on the selected drive where you want the original document to be stored. Use a trailing backslash (\). You can also click **Browse** to display the "Browse for Folder" window on which you can select the appropriate path.

Untranslated segments

You can select this option only if you selected **Create file containing untranslated segments** during analysis, which generates a file containing untranslated segments. Select this option if you want to export this file.

This exported file is in the format of an external **Translation Memory** with empty target segments. **untranslated segments automatic substitution** You can transfer this file to a machine translation system to translate these segments automatically.

Note:

machine translation system. The machine translation system must first be programmed to recognize the source and target language segments in the external format of the **Translation Memory**.

After the empty target segments have been completed by the machine translation system, you can import the file into **OpenTM2** as a **Translation Memory** and use it to translate the rest of the document. The proposals offered in the **Translation Memory** window are then preceded by an [m] (generated by **machine**).

Path

You must type the path name on the selected drive where you want the untranslated segments file to be stored. Use a trailing backslash (\). You can also click **Browse** to display the "Browse for Folder" window in which you can select the appropriate path.

Click **Export** to start document export.

Results

The document is exported in external format and stored as an HTML file in the directory under the file name you specified. The document can then be further processed with the corresponding word-processing system.

Exporting a document in external format with a path

exporting documents in external format with path **document exporting in external format with path**
You can export a document in external format with its entire path information. You use this format to distribute a translated document for checking, for example, using the relative path information of the document to build the fully qualified name for the document to be exported.

Calling sequence

To export a document in external format, select **External Format with relative path** from **Export as**. The "External Format with relative path" page is displayed (see [Figure 59](#)).

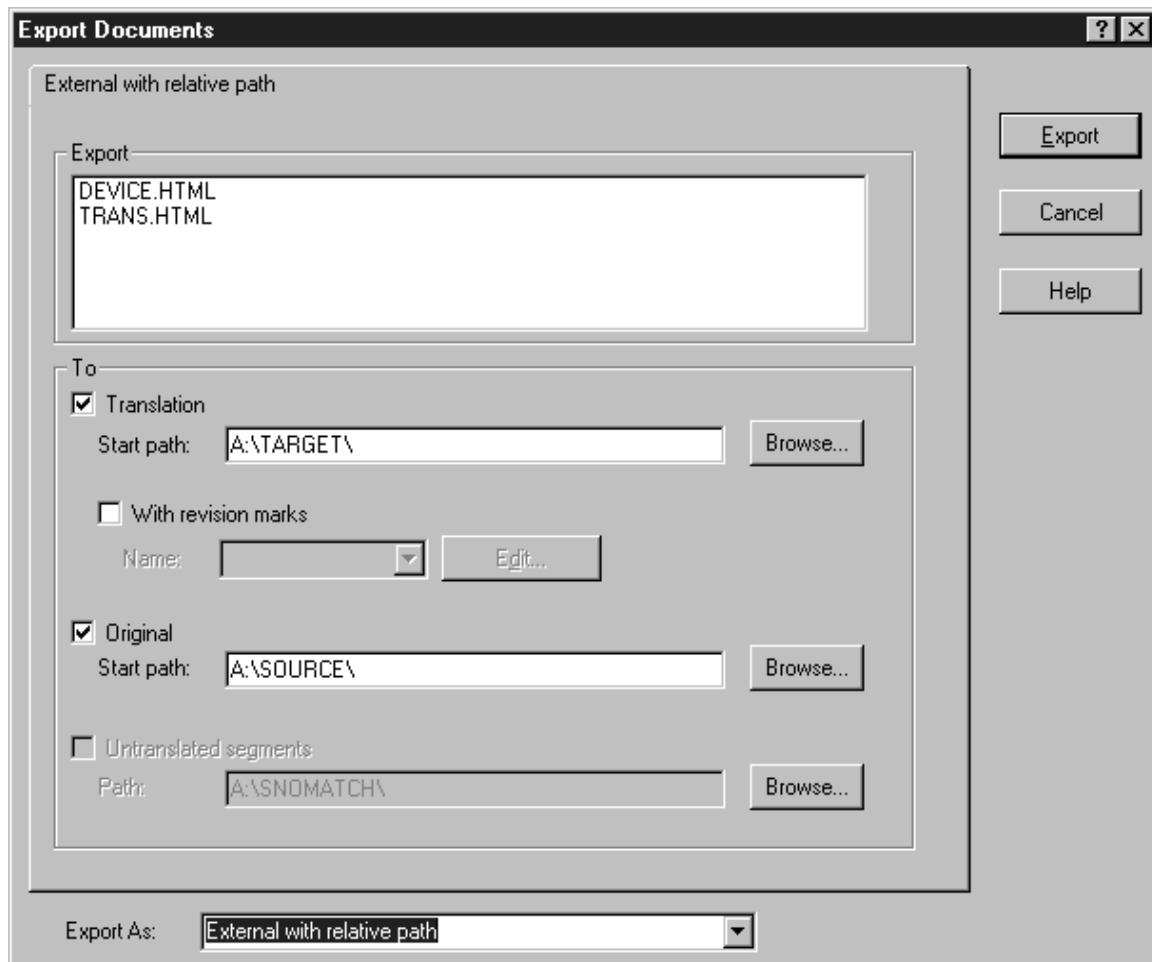


Figure 59. Export Documents window (External format with path information)

Options and parameters

The page displayed differs from the "External" page in only one field, **Start path** instead of **Path**. The following sections describe only this field. For an explanation of all other fields, refer to [Exporting a document in external format](#).

- **Start path**

Type the path name on the selected drive where you want the translated or original document to be stored. Use a trailing backslash (\). The path that you specify in this field will also be stored as part of its name. For example, if the document name is `vers\docs\device.html` and the start path is `\target\`, the document is exported under `\target\vers\docs\device.html`.

Results

The document is exported in external format under its name and path information and stored as an HTML file in the directory under the file name you specified. The document can then be further processed with the corresponding word-processing system.

If the path that you specified does not exist yet, you are asked whether you want to create the new path. If you are exporting one document, click **Yes**. If you are exporting several documents and the path is to apply to all documents, click **Yes to All**.

Exporting a document in validation format

exporting documents in validation format You can export a document in validation format for proofreading or translation validation.

Calling sequence

To export a document in validation format, select **Validation Format** from **Export as**.

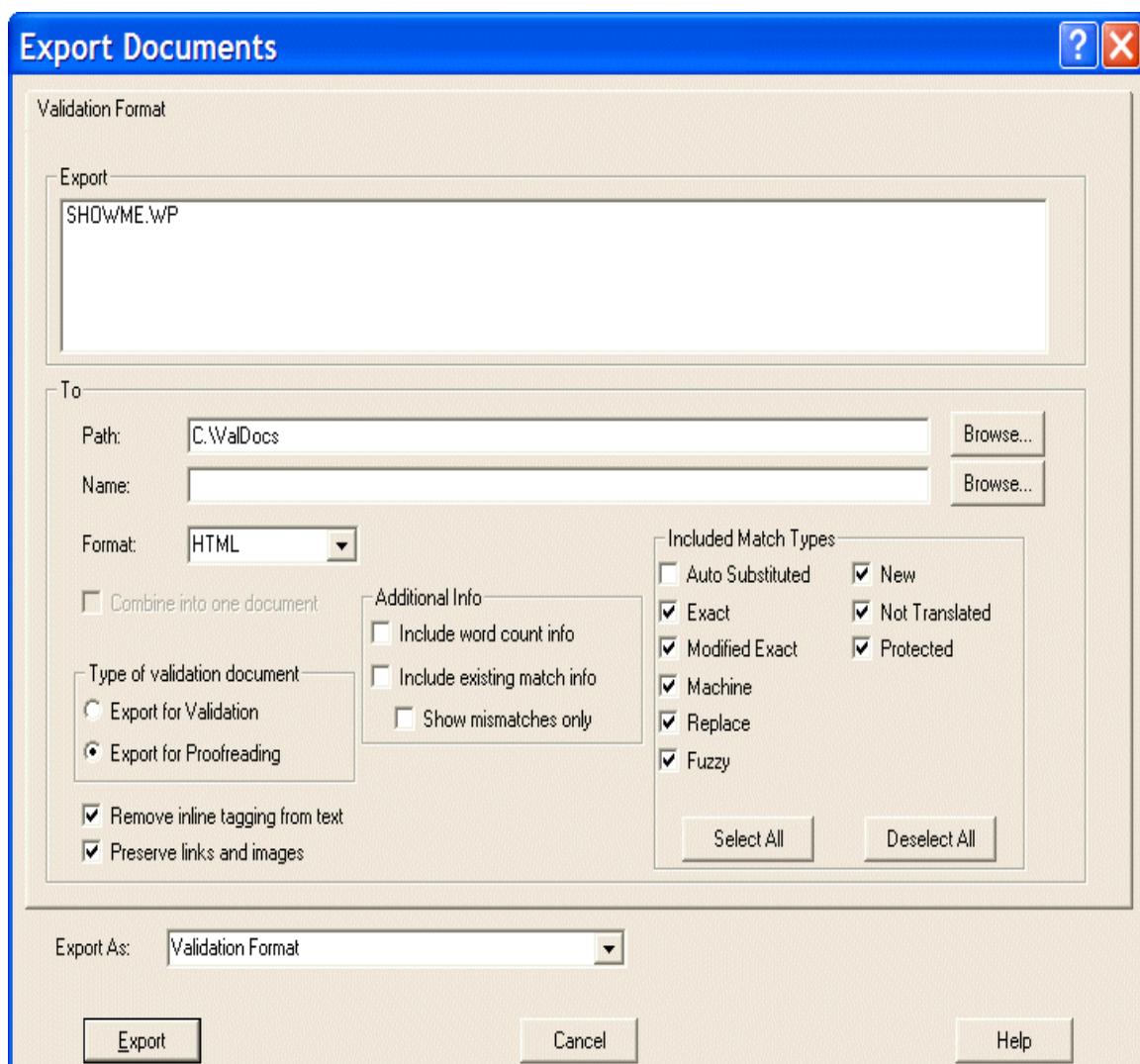


Figure 60. Export Documents window (Validation format)

Options and parameters

Path

Type the path name on the selected drive where you want the validation document to be stored. Use a trailing backslash (\). You can also click **Browse** to display the "Browse for Folder" window

on which you can select the appropriate path.

Name

Type the name of the validation document. When no name is specified, the name of the document is used as name for the validation document. In combined mode the name of the folder is used. The name can only be specified when only one document has been selected or when all documents are combined into one validation document. You can also click **Browse** to display the "Select validation document name" window on which you can select the appropriate document name.

Format

- HTML:
Use the HTML file format if you want to **display** the validation or proofreading document in the browser. No further modifications can be applied to this file format.
- XML:
Use the XML file format if you either want to display the validation or proofreading document in the browser by applying a style sheet or if you want to process the XML file using custom tools.
- DOC:
Use the DOC file format if you want to **display and edit** the validation or proofreading document in MS WORD or even in Open Office. Further processing is possible e.g. by using TRACK CHANGES or by using ADD COMMENTS. These functions are dedicated to MS WORD or Open Office.
- ODT:
Use the ODT file format if you want to **display and edit** the validation or proofreading document in Lotus Symphony or in Open Office.

Combine into one document

Select this option to combine all exported documents into one file. The name of the folder is used for the combined document. Note: this option is only active when more than one document is selected for the export.

Type of Validation Document

- Export for Validation:
This format is dedicated for **validation** purposes only. Validators request a dedicated format of the output which supports dedicated tasks and information. Please refer to the header of the generated document which contains detailed information for validators. Compared to the proofreading format users can't adjust the output by exporting single match types, the output is more static.
- Export for Proofreading:
This format is dedicated for **proofreading** purposes only. Proofreaders request a dedicated format of the output which supports dedicated tasks and information. Compared to the validation format users can adjust the output by exporting single match types.

Remove inline tagging from text

When the "Remove inline tagging from text" option is selected, all inline (neutral) tags contained in the segments are removed from the validation/proofreading document.

Preserve links and images

When the "Preserve links and images" option is selected, links in the document are left as active links within the validation document and images are converted to image links. This option works independently from the "Remove inline tagging from the text" option.

Include word count info

Check this option to add a column containing the number of source words of this segments to the proofreading document.

Include existing match info

Check this option to show information on the memory proposals which were available at the time the segment was translated.

Show mismatches only

Check this option to show only segments where the available memory proposals were not used by the translator; e.g. the segment was translated manually although an exact match existed.

Colors used in Validation Documents

- gray : protected segments
- green : segments with an extract match
- yellow : segments with fuzzy matches

- orange : segments with machine matches or replacement matches
- red : segments without matches

Document:	MatchTest.TXT			
Folder:	MatchTest			
Date created:	30.02.2006			
Project:				
Project manager:				
Validator:				
Seg #	Match	% Match Ratio	Source	Translation
1	Protected	0	A protected segment.	A protected segment.
2	Exact	100	This is an exact match.	Dies ist ein exakter Match.
3	Fuzzy	91-99	Here we have a very good fuzzy match.	Hier haben wir einen sehr guten Fuzzy-Match.
4	Fuzzy	71-90	This is a good fuzzy match.	Dies ist ein guter Fuzzy-Match.
5	Fuzzy	50-70	Now a match with a low fuzziness.	Nun ein Match mit niedriger Fuzziness.
6	Machine	100	This sentence is from the machine.	Dieser Satz ist von der Machine.
7	Replace	100	A replacement match.	Ein Replacement Match.
8	New	0	A sentence without a match.	Ein Satz ohne Match.

Figure 61. Proofreading example

Match type options:

Auto Substituted

Select this option to include segments which have been substituted automatically into the validation document.

Exact

Select this option to include segments which have been translated using an exact match into the validation document.

Modified Exact

Select this option to include segments which have been translated using an exact match which has been modified manually into the validation document.

Machine

Select this option to include segments which have been translated using a machine translation into the validation document.

Replace

Select this option to include segments which have been translated using a replace match into the validation document.

Fuzzy

Select this option to include segments which have been translated using a fuzzy match into in the validation document.

New

Select this option to include segments which have been translated manually into the validation document.

Not Translated

Select this option to include segments which have not been translated into the validation document.

Protected

Select this option to include protected (= not translatable) segments into the validation document.

Select All / Deselect All

Click **Select All** to select all match types, click **Deselect All** to deselect all match types.

Results

The document is exported as a validation document.

If the path that you specified does not exist yet, you are asked whether you want to create the new path. If you are exporting one document, click **Yes**. If you are exporting several documents and the path is to apply to all documents, click **Yes to All**.

Exporting a document in plain XML format

Exporting a document in plain XML format You can export a document in XML format for processing the segments of the document with other tools.

Calling sequence

To export a document in validation format, select **Plain XML Format** from **Export as**.

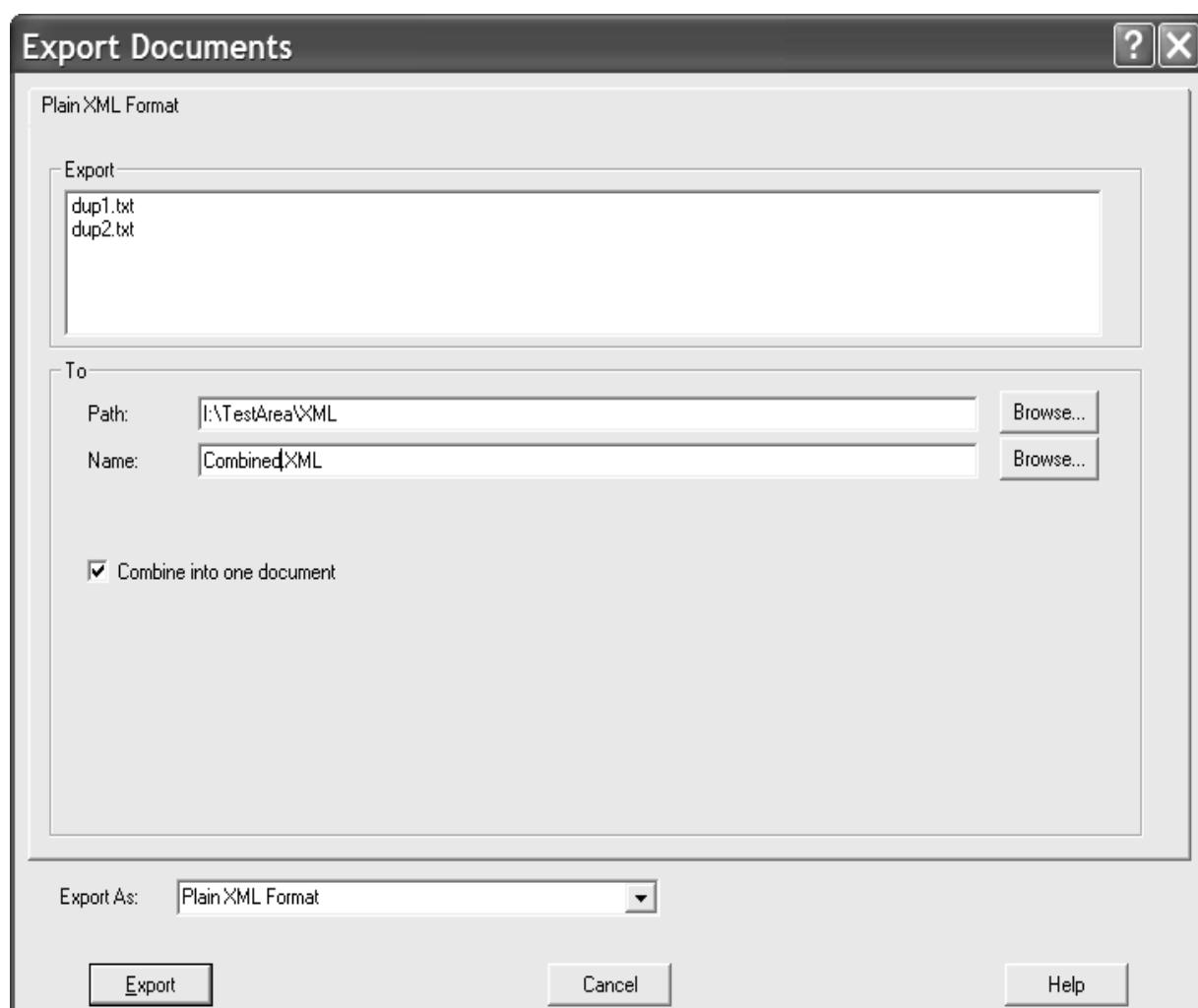


Figure 62. Export Documents window (Plain XML format)
[PIC]Figure shows Export Documents window (Plain XML format)

Options and parameters

Path

Type the path name on the selected drive where you want the XML document to be stored. Use a trailing backslash (\). You can also click **Browse** to display the "Browse for Folder" window on which you can select the appropriate path.

Name

Type the name of the XML document. When no name is specified, the name of the document is used as name for the XML document. In combined mode the name of the folder is used. The name can only be specified when only one document has been selected or when all documents are combined into one XML document. You can also click **Browse** to display the "Select XML

document name" window on which you can select the appropriate document name.

Combine into one document

Select this option to combine all exported documents into one file. The name of the folder is used for the combined document. Note: this option is only active when more than one document is selected for the export. Have been substituted automatically into the validation document.

Results

The document is exported as a XML document. If the path that you specified does not exist yet, you are asked whether you want to create the new path. If you are exporting one document, click Yes. If you are exporting several documents and the path is to apply to all documents, click Yes to All.

Exporting translated glossary terms to a Lotus^(R) Notes^(TM) database

exporting glossary to Lotus Notes document exporting glossary to Lotus Notes Lotus Notes exporting glossary to You can export translated glossary terms that are contained in a document to a Lotus^(R) Notes^(TM) database.

Calling sequence

To export the terms to Lotus^(R) Notes^(TM), select **Lotus^(R) DGW** from **Export as**.

The "Lotus^(R) DGW" page is displayed (see [Figure 63](#)).

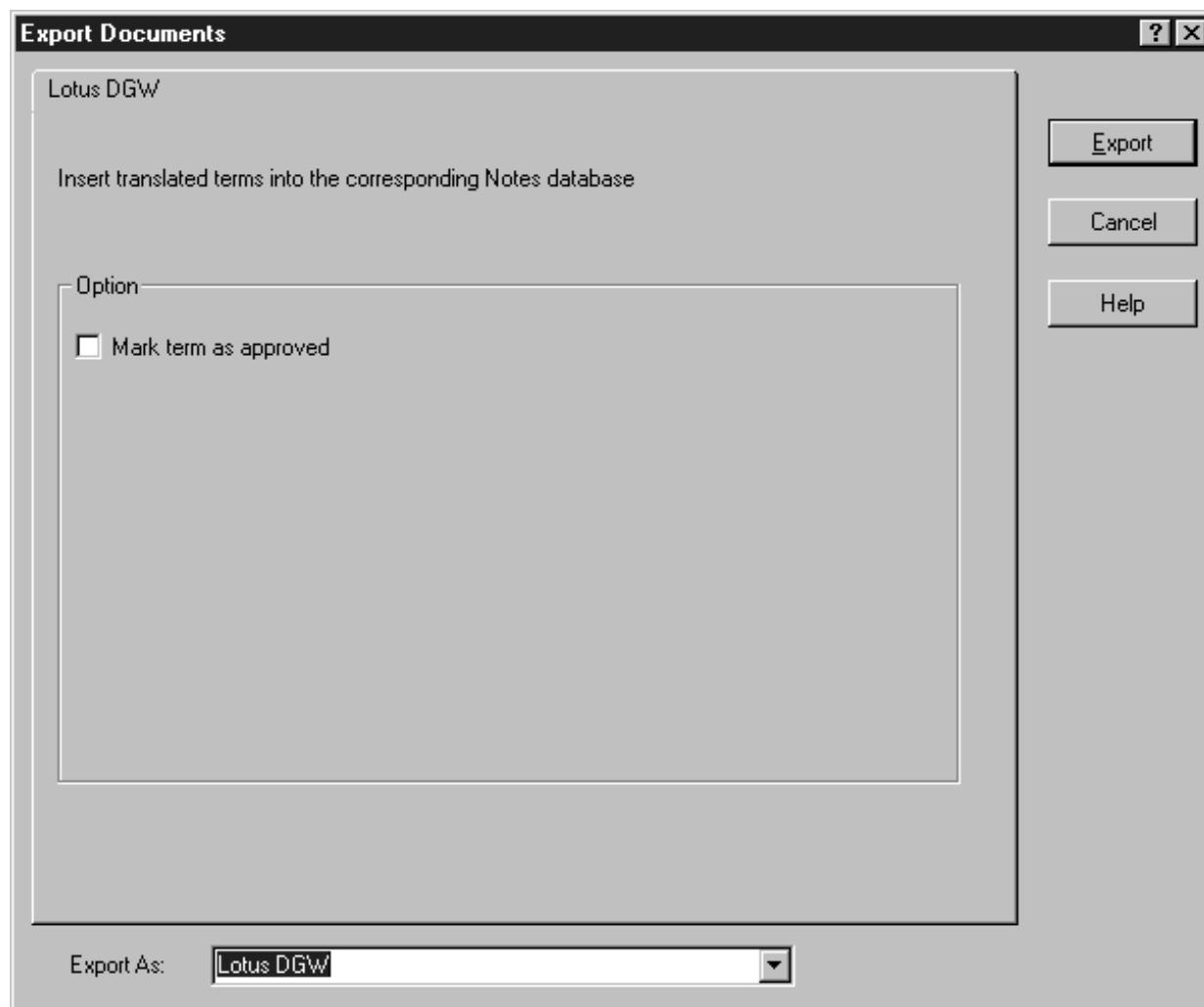


Figure 63. Export Documents window (Lotus Notes)

[PIC]Figure shows Export Documents window (Lotus Notes)

Options and parameters

Mark term as approved

Select this option if all the terms to be exported are to be marked as approved.

Click **Export** to start the export.

Results

The translated glossary terms are exported and stored in a Lotus^(R) Notes^(TM) database.

Editing document revision marks for export

revision marks, defining editing revision marks During export, OpenTM2 can insert revision marks in translated documents.

Revision marks can help reviewing a translated document. When translating an updated version of a previously translated document, many segments will remain unchanged. In most cases you can simply copy the **Translation Memory** proposals for these segments into the new document. These segments usually need no further checking because they originate from a previous translation.

The only segments that need to be reviewed are those that are translated from scratch, and those that are based on a proposal copied from the "Translation Memory" window and subsequently changed.

You can choose which segment categories you want to mark.

Prerequisites

- The "Export Documents" window must be active, and you must export the documents in external format.
- The export options **Translation** and **With revision marks** must have been selected.

Calling sequence

Click **Edit...** in the "Export Documents" window

The "Edit Revision Marks" window is displayed (see [Figure 64](#)).

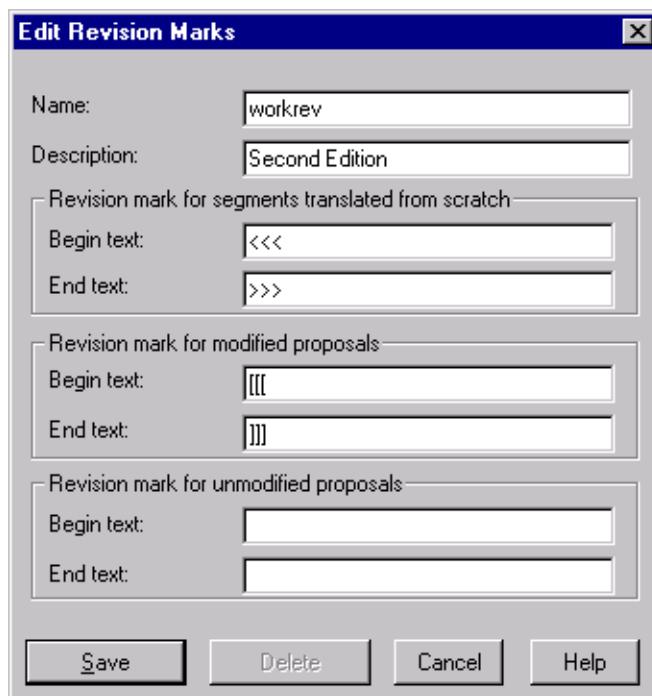


Figure 64. Edit Revision Marks window

Options and parameters

Name

If this field is still empty, type a name for the revision marks.

Description

Type a text to describe the set of revision marks; this is optional.

You can mark the following types of segments with revision marks:

- Segments that have been translated from scratch without using a proposal from the Translation Memory (... **translated from scratch**)
- Segments that have been copied from a Translation Memory proposal and then changed (... **modified proposal**)
- Segments that have been copied from a Translation Memory proposal but not changed (... **unmodified proposal**)

OpenTM2 lets you choose different symbols for marking the beginning and the end of each type of segment in the **Begin text** and **End text** fields. You can type, for example:

- Tags that produce symbols when the exported document is formatted and printed.

- Control characters that change the font type when the exported document is formatted and printed. Control characters are only recognized by certain word processors or printers.
- Eye-catching characters, such as a string of asterisks (*****).

Click **Save** to return to the "Export Documents" window.

Results

The set of revision marks is saved under the name you specified in the **Name** field. Documents exported in external format are marked accordingly. To see the result of the revision marks, browse the exported document file with a text editor or, if applicable, print the document.

Note:

When you have marked the translations with the revision mark, you are recommended not to import these documents in **OpenTM2** again, because the revision marks might be misinterpreted.

It is recommended to export the revised and finalized translation again without revision marks.

Importing a document

documentimporting importingdocuments You must import a document into **OpenTM2** before you can work with it.

A **OpenTM2** document can be in one of the following formats:

- **OpenTM2** format
This format can be received from another **OpenTM2** user who previously exported the document in **OpenTM2** format. When you import a document into **OpenTM2**, you receive the current status of both the original and translated version of the documents.
- External format (with and without relative path)
This format is produced by word-processing systems, such as WordPerfect or Microsoft^(R) Word for Windows^(R).
When you import a document in external format you can decide whether you want to import it either under its name only or under its name and path information. Name and path uniquely identify a document in a folder. If you import a document under its name and path, you can have several documents in a folder with the same name, provided that the path differs.

In addition, you can extract glossary terms from a Lotus^(R) Notes^(TM) database and import them as a document into **OpenTM2**.

Prerequisites

The active window is the "Document List" window of the folder into which the document is to be imported.

Calling sequence

Select **Import...** from the **File** menu. The "Import Documents" window is displayed (see [Figure 65](#)).

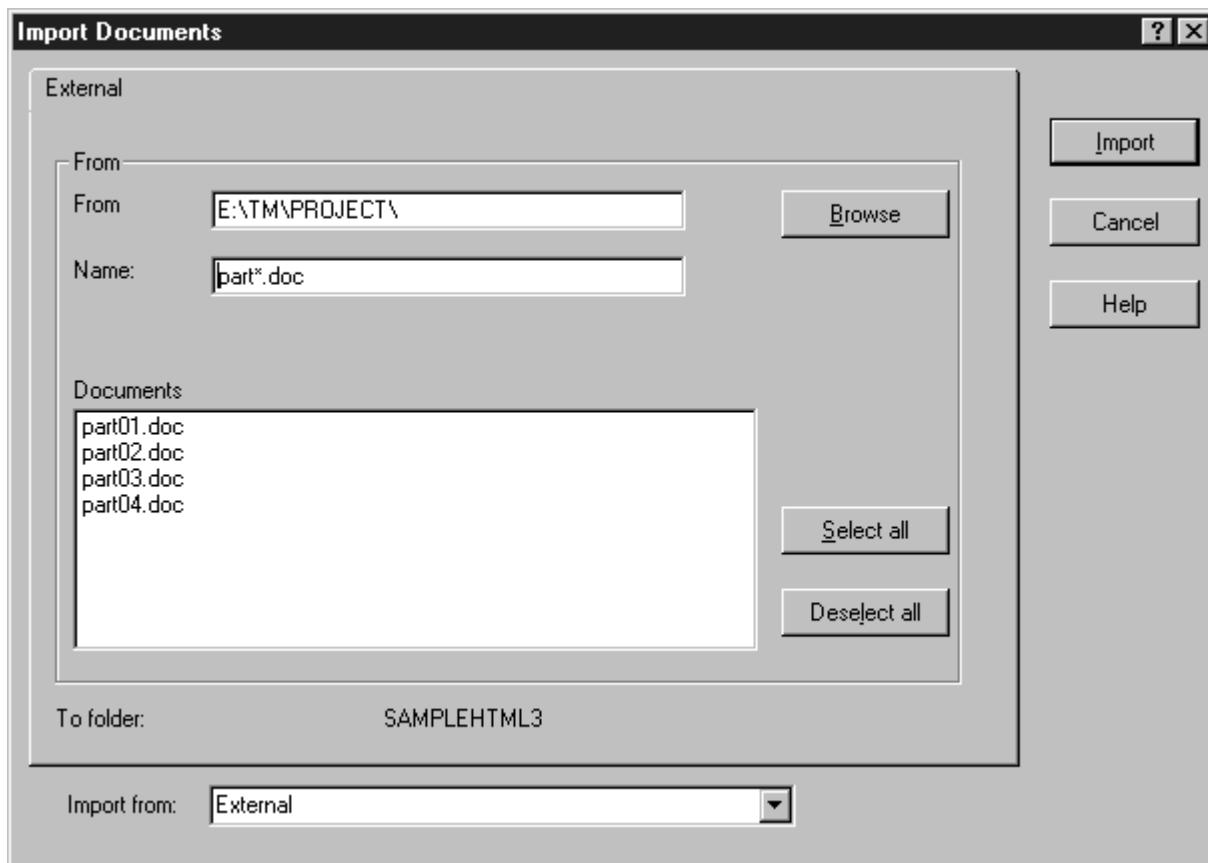


Figure 65. Import Documents window
[\[PIC\]](#)Figure shows Import Documents window

Importing a document in OpenTM2 format

Calling sequence

To import a document that is in OpenTM2 format, select **OpenTM2 Format** from **Import from**. The "OpenTM2 Format" page is displayed (see [Figure 66](#)).

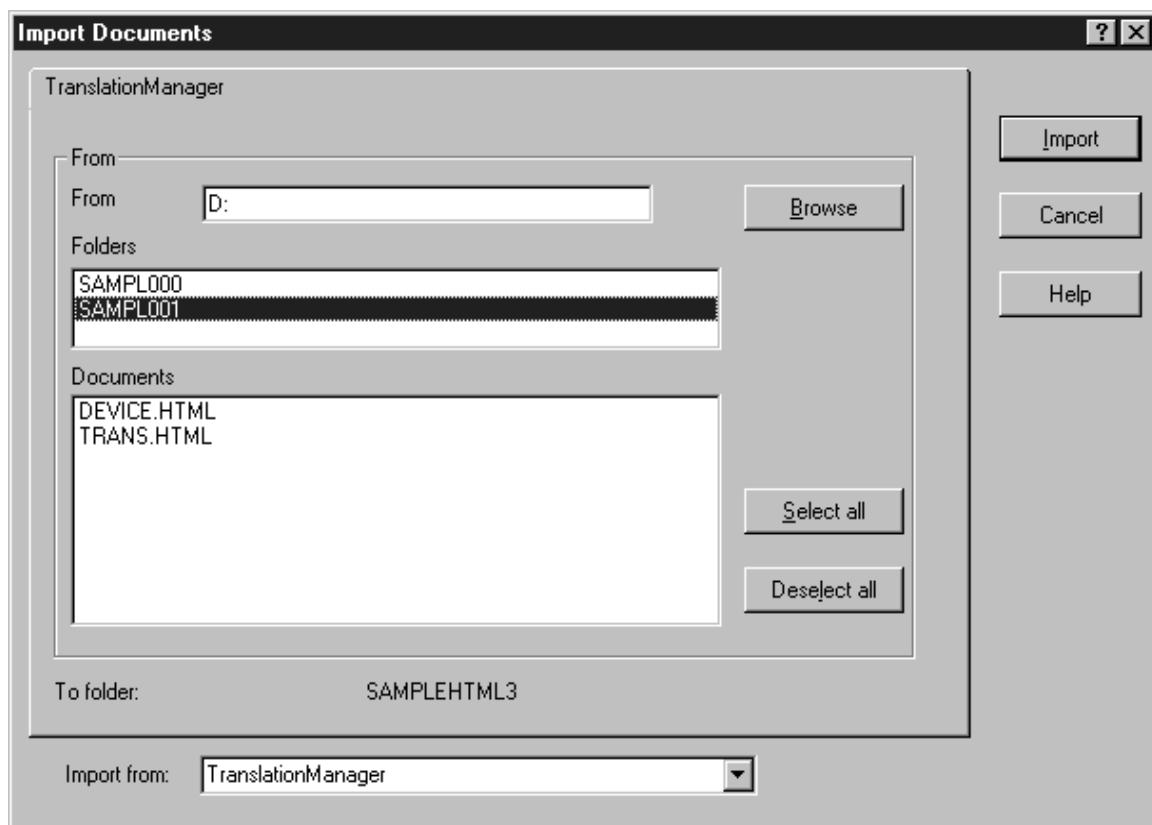


Figure 66. Import Documents window (OpenTM2 format)

Options and parameters

From

To specify the current location of the documents:

- Specify the drive where the documents to be imported currently reside. You can also click **Browse** to display the "Browse for Folder" window on which you can select the appropriate drive.
- Double-click the folder containing the documents to be imported, in the **Folders** list.
- Click one or more document names in the **Documents** list.

To import all documents listed in the **Documents** list, click **Select all**.

To start a new selection of documents, click **Deselect all**.

To folder

Shows the name of the folder to which you want to import the documents.

All files contained in the selected folder are listed in **Documents**. Select the documents you want to import, then click **Import** to start the import procedure.

Results

The selected documents are imported and you can start analyzing and translating.

Importing a document in external format

Calling sequence

To import a document produced by word-processing systems, select **External Format** from **Import from**.

The "External Format" page is displayed (see [Figure 1](#)).

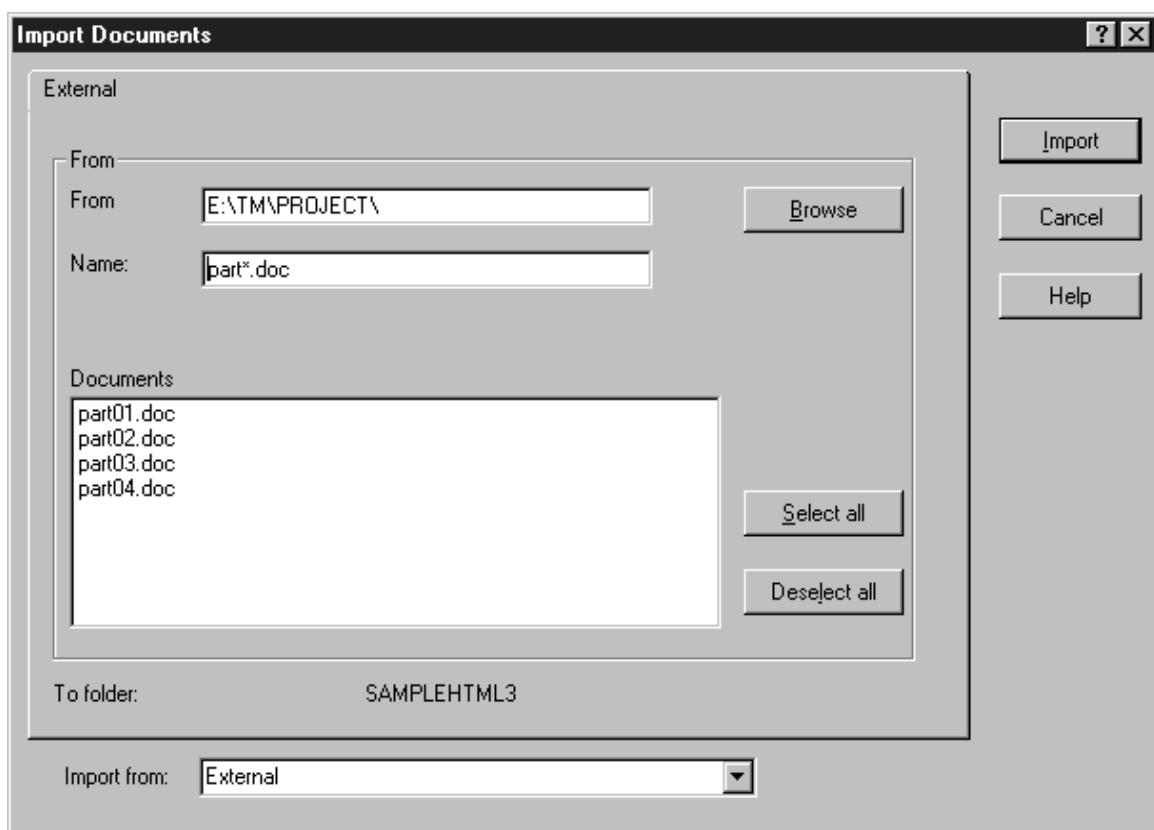


Figure 67. Import Documents window (External format)

Options and parameters

From

To specify the current location of the documents:

- Specify the path in which the documents to be imported currently reside. You can also click **Browse** to display the "Browse for Folder" window on which you can select the appropriate path.
- Type the **Name** of the document to be imported. You can use global file name characters in this field as follows:
Type only the common part of the file name in the **Name** field to display a subset of the available files in the **Files** list. Then add an asterisk (*) to represent the part of the file names that is different. For example, to import the files part01.doc, part02.doc, part03.doc, and part04.doc, type part*.doc in the **Name** field and press Enter.
- The **Documents** list displays the files that match the specification you have entered in the

Name field. Select the files you want to import as documents.

To import all documents listed in the **Documents** list box, click **Select all**.

To start a new selection of documents, click **Deselect all**.

To folder

Shows the name of the folder to which you want to import the documents.

All selected files are listed in **Documents**. Select the documents you want to import, then click **Import** to start the import procedure.

The "Document Properties" window is displayed for the first document (see [Figure 68](#)).

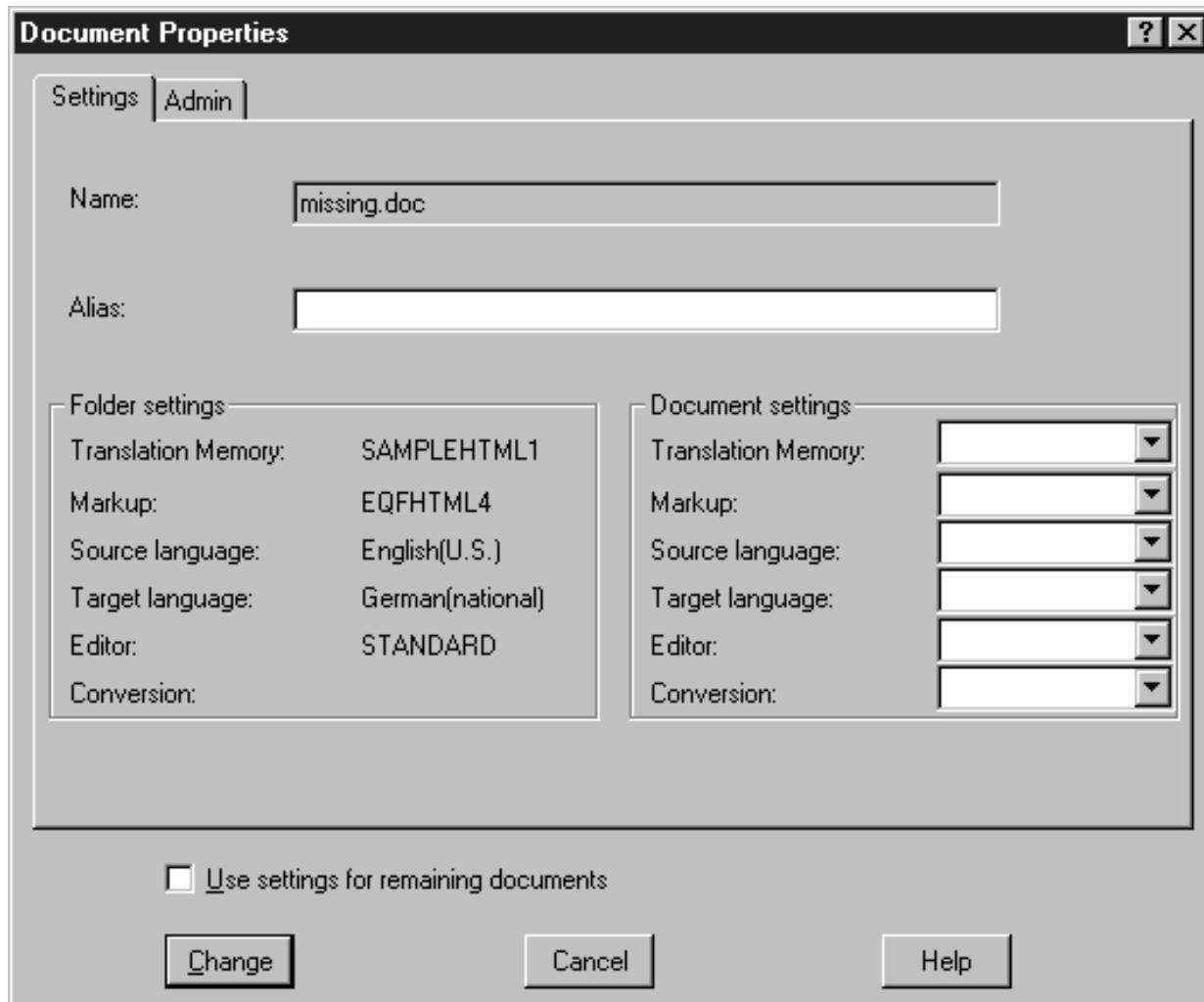


Figure 68. Document Properties window
[\[PIC\]](#)Figure shows Document Properties window

Options and parameters

If you import documents with the same properties as the folder, you do not specify anything in this window. Click **Change** to import the document.

If several documents are selected for importing, you must specify the properties for each document to be imported.

On the "Settings" page, specify:

Alias

An alias name for the document. Such a name is useful if you want to get the same exact matches from the **Translation Memory** as for another document that you already translated. You then assign the name of the other document as alias name for this document. Assume, for example, that document docupdate is an update of docnew and you want to get the same exact matches from the **Translation Memory** that docnew received. In this case, you would assign docnew as the alias name to docupdate.

If an alias name exists, **OpenTM2** uses this name instead of the actual document name to access the correct **Translation Memory**.

The same alias name can be assigned to several documents. An alias name can have up to 256 characters.

Folder settings

In this box, the properties are displayed that are already defined for the folder in general:

Translation Memory

The name of an existing **Translation Memory**

Markup

The name of an existing markup table

Source language

The name of a supported source language

Target language

The name of a supported target language

Editor

The editor used for the documents in the folder

Conversion

The code page used for the documents in the folder

Document settings

Each document can have its own properties and can be different from those of the folder. If you import a document for the first time, the boxes in this group are empty. If you import a document that already exists in the folder, the properties that you specified for the first import and that differ from the folder properties are displayed as default values, which you can change. If the document has the same properties as the folder, you do not have to enter anything, and the document *inherits* the folder properties. Document properties that are different from the folder properties can be selected from the respective list boxes:

Translation Memory

The name of an existing **Translation Memory**

Markup

The name of an existing markup table

Source language

The name of a supported source language

Target language

The name of a supported target language

Editor

The editor to be used for the imported document

Conversion

The code page to be used for the imported document

Use settings for remaining documents

This option is only displayed if you want to import more than one document. If you want to assign the same properties to all documents, select this option.

For information on the "Admin" page, refer to [Admin page](#).

To save the specified properties, click **Change**.

Results

The selected documents are imported and you can start analyzing and translating.

Importing a document in external format with a path

You can import a document in external format with its entire path information. You can then have several documents in a folder that have the same name provided that the path differs.

Calling sequence

To import a document produced by word-processing systems, select **External Format with**

relative path from **Import from**. The "External Format with relative path" page is displayed (see [Figure 69](#)).

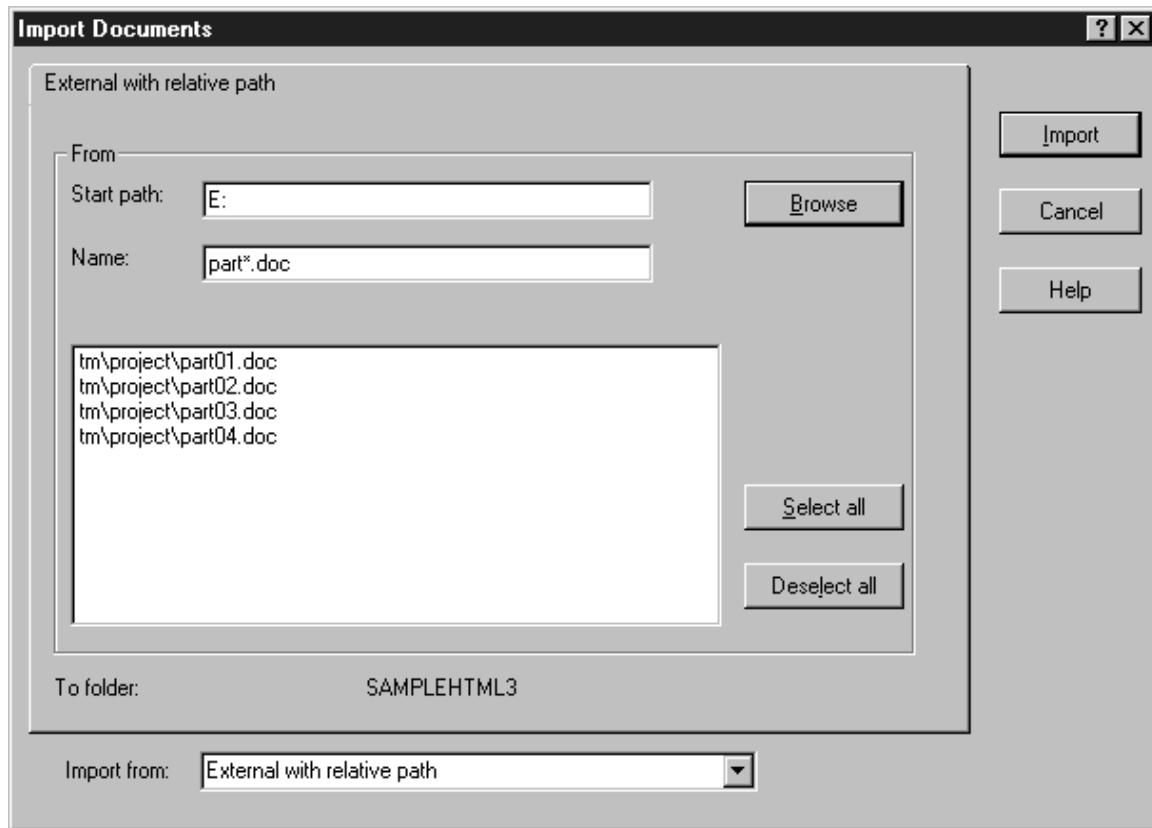


Figure 69. Import Documents window (External format with path information)

Options and parameters

Start path

Specify the drive on which the documents to be imported reside or specify the entire path in which they are stored. What you specify depends on which path information you want to become part of the documents' name. If you do not want the full path information become part of the documents' name, type the directory to be omitted in this field. For example, if you do not want directory tm to be part of the documents' name, type TM in the **Start path** field. Document part01.doc is then imported under the name project\part01.doc.

You can also select the path from the "Browse for Folder" window after clicking **Browse**.

Name

Type the name of the document to be imported. You can use global file name characters in this field as follows:

Type only the common part of the file name to display a subset of the available files in the files list. Then add an asterisk (*) to represent the part of the file names that is different. For example, to import the files part01.doc, part02.doc, part03.doc, and part04.doc, type part*.doc in the **Name** field and then press Enter. The files list then lists all documents whose name matches your specifications. The name of the documents listed also contains the path information that you did not specify in the **Start path** field and therefore becomes part of the documents' name.

Files

Displays the files that match the specifications you entered in the **Start path** and **Name** fields.

To folder

Shows the name of the folder to which you want to import the documents.

Select the documents you want to import from the files list. To import all documents listed in the files list, click **Select all**. To start a new selection of documents, click **Deselect all**. Click **Import** to start the import procedure.

The "Document Properties" window is displayed for the first document (see [Figure 2](#)). For more information about the contents of this window, refer to [Importing a document in external format](#).

Results

The selected documents are imported. If you left the **Start path** field empty, the documents are imported under their name and full path information. If you specified a path in the **Start path** field,

the path information of the imported documents excludes the specified path.
You can now start analyzing and translating the imported documents.

Importing glossary terms from a Lotus^(R) Notes^(TM) database

You can import the glossary terms of a Lotus^(R) Notes^(TM) database into OpenTM2. The name of the document's markup table is always LOTUSNGD.

Calling sequence

To import the glossary terms from Lotus^(R) Notes^(TM), select **Lotus^(R) DGW** from **Import from**.

The "Lotus^(R) DGW" page is displayed (see [Figure 70](#)).

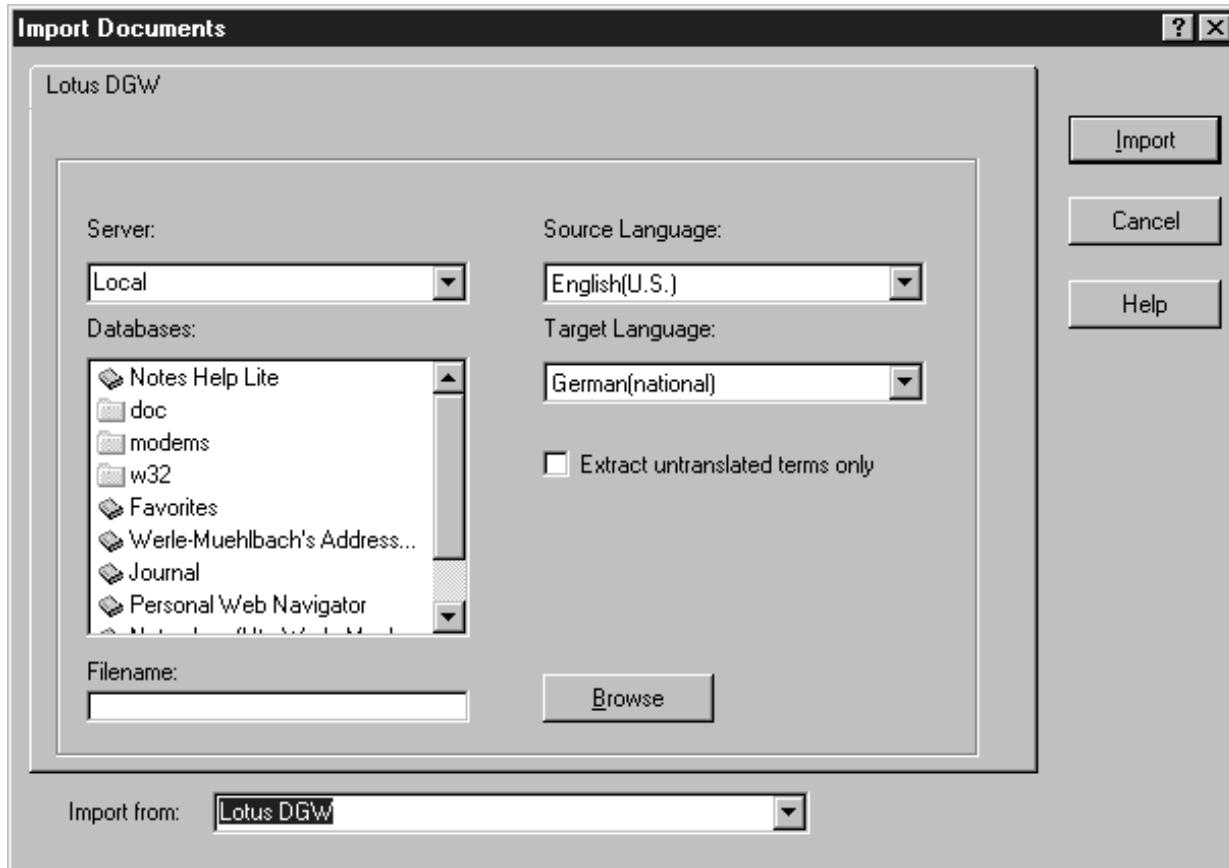


Figure 70. Import Documents window (Lotus Notes)
[\[PIC\]](#)Figure shows Import Documents window (Lotus Notes)

Options and parameters

Server

Select or specify the server on which the database containing the glossary terms is located.

Databases

Displays all databases available on the selected server. Click a database to select it.

File name

Displays the database that you selected in **Databases**.

Source language

Select the source language of the glossary terms.

Target language

Select the target language of the glossary terms.

Extract untranslated terms only

Select this option if you want to import only the terms that have not been translated yet.

Click **Import** to start the import.

Results

The glossary terms are extracted from the specified Lotus^(R) Notes^(TM) database and are stored in a document in the SOURCE directory of the target folder. The name of the document is equal to

that of the Lotus^(R) Notes^(TM) database.

Copying and moving documents between folders

deletingdocuments documentcopying copyingdocuments between folders movingdocuments
between folders You can copy or move one or more documents in a folder to another folder. This is an alternative to exporting and then importing documents, and it is recommended if you deal with a large number of documents. The advantage of this method is that you use the Windows-like multiple-select and drag-and-drop techniques to copy or move documents. Nevertheless, internally OpenTM2 exports the selected documents to a temporary directory and imports them into the target folder. OpenTM2 maintains the history log files of the source and target folders, as described in [Results.history log filecopying documents moving documents](#)

Prerequisites

The source folder containing at least one document must be open, and the corresponding "Document List" window must be shown. The "Document List" window with the target folder must be open. You can also have your folder list displayed in the Explorer-like tree view. Make sure that the target folder is visible.

Calling sequence

1. Select one or more documents in the "Document List" window that you want to copy or move. Do this by holding down the Ctrl key and clicking the documents. To select a series of documents, use the Alt key respectively.
2. If you want to drag and drop the selected documents to the target folder:
 - To move the documents, drag them to the target folder. Note the small drag icon.
 - To copy the documents, press SHIFT and drag them to the target folder. Note the small plus symbol beside the drag icon, indicating a copy operation.
3. If you want to use the move/copy menu:
 - With the documents still selected, click right on the document.
 - Open the target folder
 - Click **PASTE**
 - Select document list window
 - Right mouse click on document list window
 - Click **Copy** respectively **Move** (The button corresponds to your previous selection.)

Results

The selected documents are copied or moved to the selected target folder.

The history log files of the source and target folder are updated as follows:

- The history log entries of the selected documents from the source folder are merged into the target folder's history log. They are not deleted from the source folder's history log.
- The target folder's history log gets an entry that indicates that the documents were copied, respectively moved, with the "copy/move" function.
- The source folder's history log only gets an entry when documents were moved, indicating this fact.

Opening a document

startingtranslations documentopening openingdocuments You must open a document before you can start translating.

If the selected document has not been analyzed, OpenTM2 automatically analyzes the document based on the properties of the document and the folder. For more information on the analysis, see [Analyzing a document](#).

Prerequisites

The "Document List" window of the respective folder is active.

Calling sequence

Select:

1. The document from the "Document List" window
2. **Open** from the **File** menu

Alternatively, double-click the document in the "Document List" window.

You are taken to the **Translation Environment**, where the selected document is contained in the "Translation" window.

Results

The document is segmented and you can start translating it.

To leave the **Translation Environment**, you must close the document.

Searching and replacing text in documents

OpenTM2 provides a Global Find and Replace function, which works across all or some documents of a folder.

This function might be helpful if you are looking for a particular piece of text of which source document you are not sure . Also, you may need to change a particular wording, but do not want to type the new text manually at each occurrence. OpenTM2 can search through a series of documents to locate the required text and change it automatically. Any change is automatically reflected in the associated Translation Memory to be available for future translations.

Note that OpenTM2 searches and replaces only pieces of text that can be translated. Markup tags cannot be changed. You can control whether OpenTM2 searches in protected text areas, like inline tags, for translatable text. This is helpful if you want to include markup tag attribute values in the search process.

Prerequisites

The folder must exist, and its documents must have been analyzed.

Calling sequence

You can start this function from a "Folder List" window or from a "Document List" window. Here the start from the "Document List" window is described.

Select:

1. One or more documents you want to have searched through from the "Document List" window
2. **Global Find and Replace...** from the **File menu**

The "Global Find and Replace" window is displayed (see [Figure 71](#)).

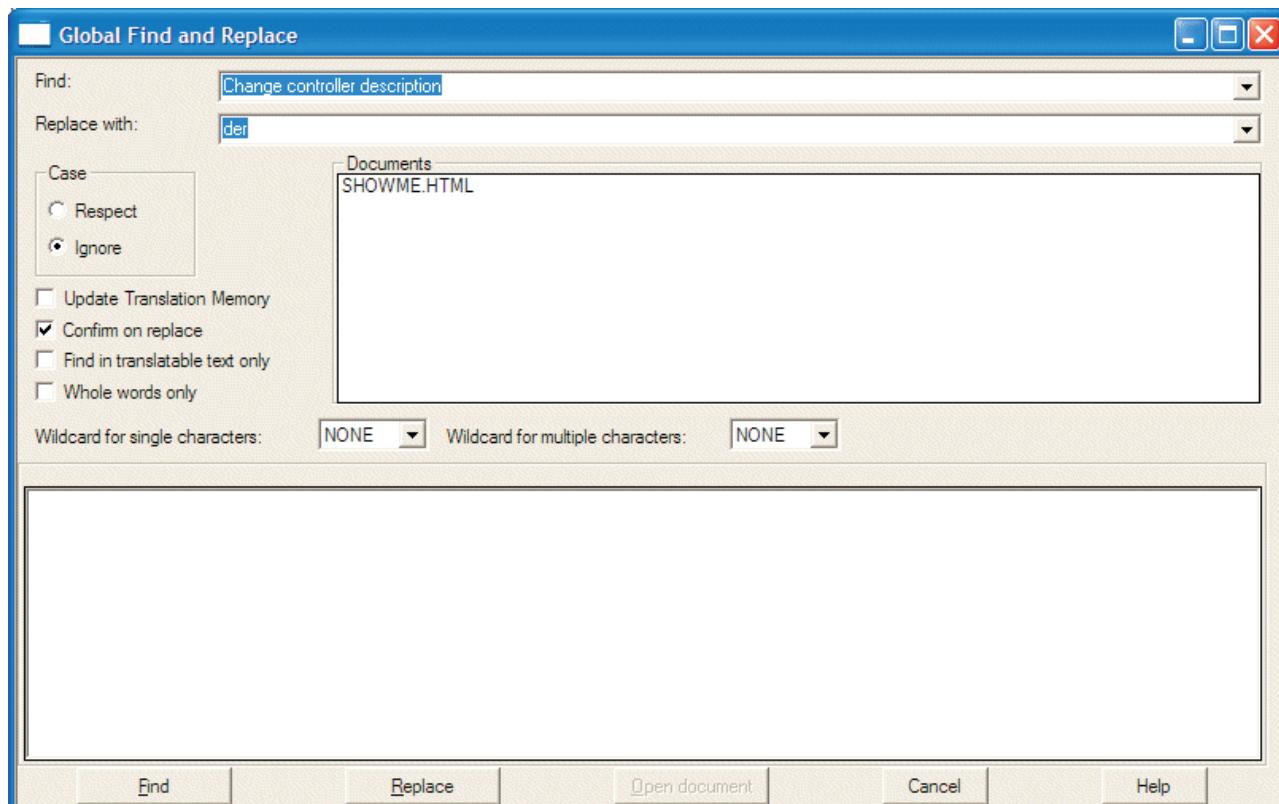


Figure 71. Global Find and Replace window
[\[PIC\]](#)Figure shows Global Find and Replace window

Options and parameters

Find

the text for which you are looking. You can also select one of the last five terms that you searched for.

If you are searching for 2 or more terms combined with AND, all terms must be contained in the same segment to form a match.

If you are searching for several terms combined with OR, either one of the terms must be contained in the same segment to form a match.

If a search term is negated with the NOT operator, a segment will only match if the negated term is not contained in the segment.

Note:

You can use wildcards. * is the placeholder for multiple characters, ? is the placeholder for a single character.

Note:

If a search string contains AND / OR / NOT enclose the search string between double quotes. If you want to search terms containing double quotes the search term has to be enclosed in double quotes and the double quote being searched has to be duplicated. e.g. if you want to search for the term "Null" ("Null" followed by a double quote) the term has to be entered as "Null"" in the find field.

Replace with

To change the term you are searching for to another term, type the term with which you want to replace the term that you specified in the **Find** box. You can also select one of your last five replacements.

Note:

You can only replace terms in translated segments.

Case

Specify whether the search should **respect** or **ignore** case sensitivity.

Documents

Displays the list of documents that are to be searched.

Update Translation Memory

Select this box if you want your changes to be reflected in the Translation Memory. Any changed text segment is then updated in the Translation Memory associated to the folder.

Confirm on replace

Select this check box if you want OpenTM2 to stop before a text is changed. You are then prompted to confirm the change.

Find in translatable text only

Select this check box if you want OpenTM2 to search the text you are looking for in translatable text only.

If this check box is not selected, OpenTM2 searches also in protected text areas, like inline tags, for translatable text. This process might locate markup tag attribute values.

Whole words only

Select this check box if you want to find whole words only, not occurrences where the Find string is part of a word.

If this check box is not selected, the Find term is also found if it is part of a word

Wildcard for single characters

Select here the character to be used as wildcard for single characters or de-activate the single character wildcard by selecting "None". When used in the search string the wildcard character matches any character of the document.

Example: assuming that "?" has been selected as wildcard for single characters, the search string "f?r" will match the following words: "fur", "for" but not the words "father", "feather", "fr"

Wildcard for multiple characters

Select here the character to be used as wildcard for multiple characters or de-activate the multiple character wildcard by selecting "None". When used in the search string the wildcard character matches none, one or more characters in the document. Example: assuming that "*" has been selected as wildcard for multiple characters, the search string "f*r" will match the following words: "four", "father", "feather", "fr", "for"

Display box

This area shows a part of the document containing the found piece of text. The found part is highlighted. You are also informed about the name of the document and the number of the segment containing the piece of text.

For easier perception of found text the lines are wrapped if they exceed the available width. You can enlarge the text area by enlarging the "Global Find and Replace" window. This retains the dialog control elements, but dynamically resizes the available window space for the text area.

Find

Starts the search for the term that you specified in the **Find** box in the first document listed and displays a small portion of the document in which the first occurrence of the term is found, in the text area. The term found is highlighted. You can then select either of the following:

- **Open document** to view the entire document in which the term was found.
- **Find** to move to the next occurrence of the term, which can be in the same or in another document to be searched. Select this button until you have reached the end of the last document to be searched or you want to stop searching.

Replace

If you selected the **Confirm on change** check box, a small portion of the document in which the first occurrence of the term is found is displayed in the display box and the found term is highlighted. You can then select one of the following:

- **Open document** to view the entire document in which the term was found.
- **Find** to move to the next occurrence of the term without changing the term. The next occurrence can be in the same or in another document to be searched.

If you did not select the **Confirm on change** check box, all occurrences of the term are replaced without you being asked to confirm the changes.

If you selected the **Update Translation Memory** check box, changes to translated segments are reflected in the Translation Memory associated with the folder.

Cancel

Stops the search and the interaction with this window and returns you to the starting point. Any changes you have made are not saved.

Help

Gives information to help you interact with this window.

Results

The specified piece of text is located at every occurrence within a document and changed where necessary. The associated Translation Memory is updated if requested.

Examples

"and"

Finds all occurrences of the string **and**. The double quotes around the term are required as **and** is a logical operator

IBM and not SAP

Finds all occurrences of the string **IBM** in segments which do not contain the term **SAP**

f?r

Find all occurrences of strings starting with **f** and ending with **r**, e.g. for, fur, futur, fair, far

f?r

Find all occurrences of strings starting with **f** followed by one other character followed and ending with **r**, e.g. for, far, fur

"one"***

finds all occurrences of the term **one** followed by a double quote

finds all occurrences of a single double quote

Search segments with fuzzy matches

OpenTM2 provides a Search Segments with Fuzzy Matches function, which works across all or some documents of a folder or across a group of folders.

This function might be helpful in the pre-translation process to identify segments which can be easily translated by using a fuzzy memory proposal and change some tagging or a few words.

Note that OpenTM2 searches and replaces only pieces of text that can be translated. Markup tags cannot be changed. You can control whether OpenTM2 searches in protected text areas, like inline tags, for translatable text. This is helpful if you want to include markup tag attribute values in the search process.

Prerequisites

The folder must exist, and its documents must have been analyzed.

Calling sequence

You can start this function from a "Folder List" window or from a "Document List" window. Here the start from the "Document List" window is described.

Select:

1. One or more documents you want to have searched through from the "Document List" window
2. from the menu

The "Search Segments With Fuzzy matches" window is displayed (see [Search segments with fuzzy matches](#)).

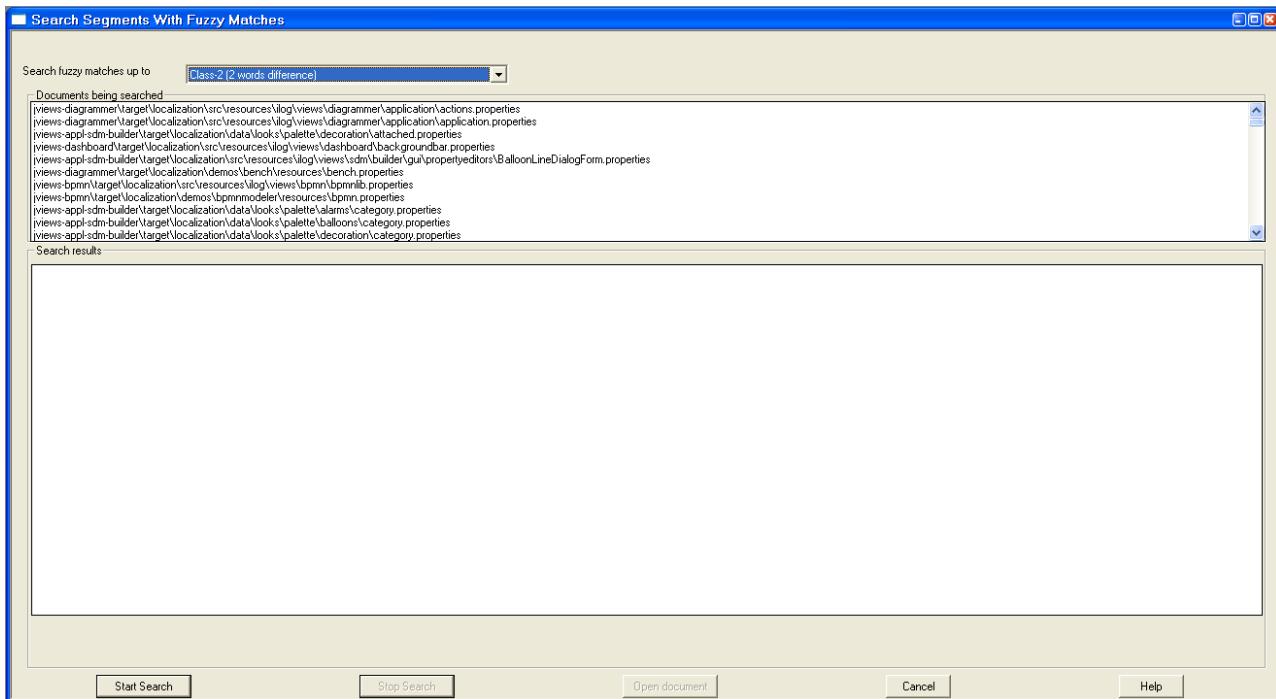


Figure 72. Search Segments With Fuzzy matches window
[PIC]Figure shows Search Segments With Fuzzy matches window

Options and parameters

Search fuzzy matches up to

select here the class of the segments with fuzzy matches being searched.

The class is the number of different words between the segment and the memory proposal.

Differences in white spaces and inline tagging are ignored.

Search results

In this area the results of the search are displayed. For each entry the folder name, the document name, the segment number, the segment data and the source of the memory proposal is shown. The differences between the segment text and the memory proposal are highlighted.

Start search

Starts the search for segments with fuzzy matches. The segments which match the selected class are displayed in the result area.

Stop search

Stops the current search.

Open document

Opens the document which contains the selected segment and makes the segment the active one. Opens the document which contains the selected segment and makes the segment the active one. You can also open the document and position to the segment by double-clicking the entry in the result list.

Cancel

Stops the search and the interaction with this window and returns you to the starting point. Help Gives information to help you interact with this window.

Results

The segments matching the selected class are displayed in the result area. Results displayed in the Search Segments With Fuzzy matches window (see [Search segments with fuzzy matches](#)).

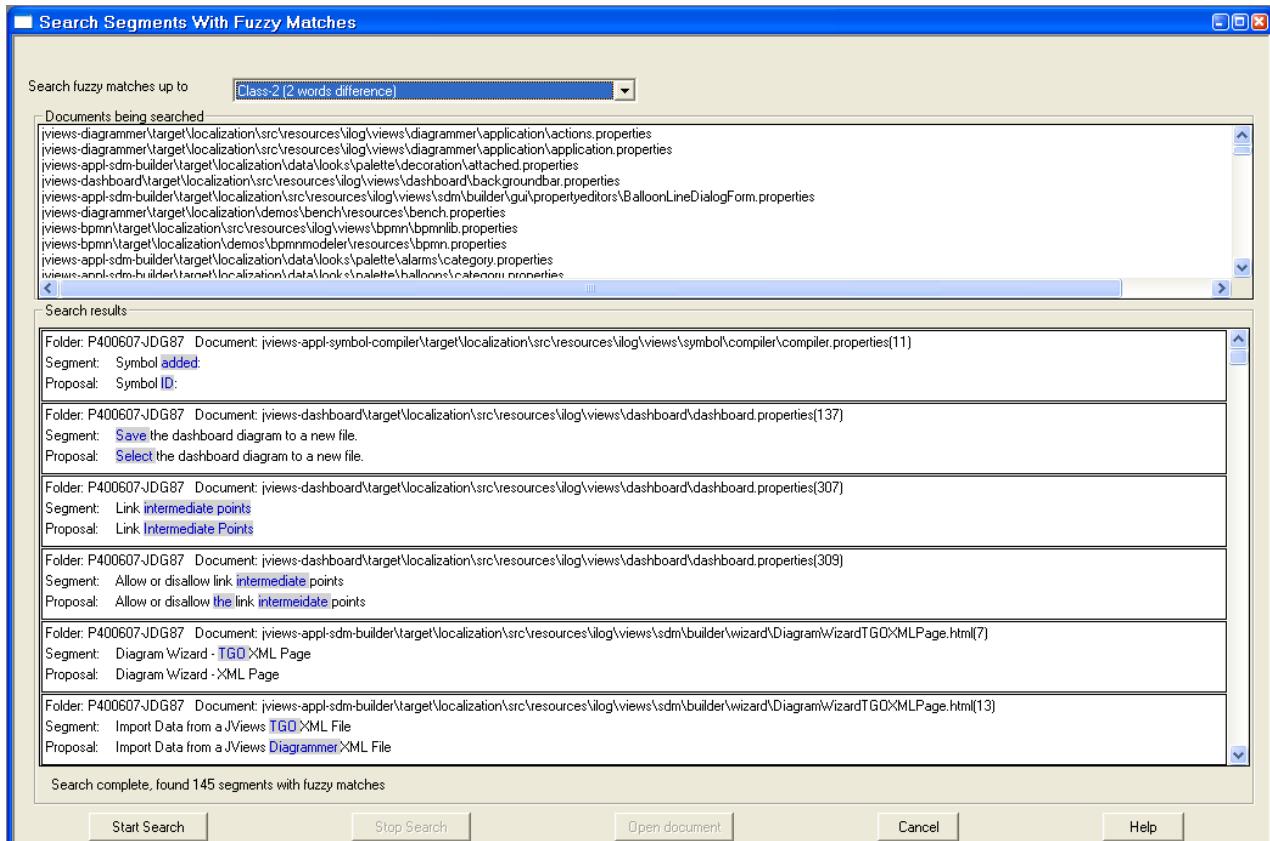


Figure 73. Search Segments With Fuzzy matches window
[PIC]Search Segments With Fuzzy matches window

Spellchecking a document

documentchecking their spelling spellcheckingdocuments A translated document can contain typing errors or misspelled words. **OpenTM2** offers a spellchecking function you can use at any time during translation.

Spellchecking is supported for all languages for which language support is installed. For each misspelled word, **OpenTM2** shows a correction proposal, which you can accept, overwrite, or skip.

Note that only the spelling of translated segments is checked.

Prerequisites

The language support file for the target language of the document must be installed. No spellchecking is supported for the following languages: Romanian, Slovene, Croatian, Slovakian, Lithuania, Latvia, Estonia.

Calling sequence

If you are not already in the **Translation Environment**, start it by selecting:

1. A document from the "Document List" window
2. **Open** from the **File** menu
3. The "Translation" window
4. **File...** from the **Spellcheck** menu

The Spellcheck function starts immediately at the beginning of the translated document, and the "Spellcheck" window (see [Figure 1](#)) is displayed.

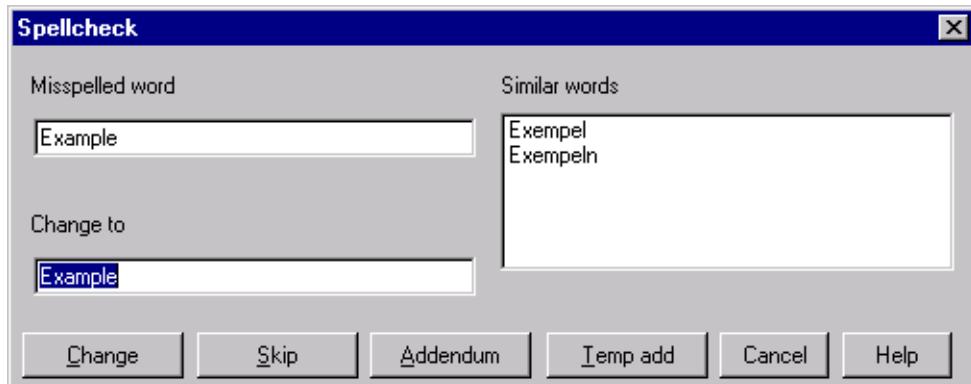


Figure 74. Spellcheck window

Options and parameters

Misspelled word

The first word in the document that is misspelled or not in the language support file.

Similar Words

A list of correctly spelled alternatives for the misspelled word. Double-click the correctly spelled word to select it. Or, click **Skip** to continue with the next misspelled word without selecting any of the alternatives. For some words, this list may be empty.

Change to

The word selected as the correctly spelled one. You can overtype it if you need a new spelling.

You can accept a spelling in several ways. Select:

Change

To accept the spelling in the **Change to** field.

Addendum

To add a new spelling to the target language support file. Then this spelling will not be shown as misspelled anymore.

Temp Add

To add the new spelling temporarily to the target language support file. When you leave **OpenTM2**, this new spelling is not saved.

When all misspelled words have been either changed or skipped, a message appears telling you that spellchecking is complete.

Results

The corrections you accepted are substituted for the misspelled words in both the document and the **Translation Memory**.

Editing the spellcheck addendum

spellcheckingaddendum terms, editing You can add terms which have been indicated as misspelled by the spellcheck facility (although spelled correctly) to an addendum for the language-support file of the current target language. During the next spellcheck, terms of this addendum are considered to be correct. You can modify this addendum at any time, for example, to remove terms that have been added by mistake.

Prerequisites

The document must be translated, and you must be in the Translation Environment.

Calling sequence

Select **Edit addendum terms...** from the **Spellcheck** menu.

The "Edit Addendum Terms" window is displayed.

Options and parameters

The terms in the addendum can be modified as follows:

- To change a term, overtype it.
- To add a term, press Enter, and type the new term.

- To remove a term, mark it and press Enter.

Clipboard operations (cut, copy, paste) are supported. To save your changes, select **Save**. To leave the window without saving your changes, select **Cancel**.

Results

The corrections you made are saved and taken into account during subsequent spellchecks.

Translating a document

translatingdocuments documenttranslating To prepare the translation of documents you can generate terminology lists, for example, to add new terms to a dictionary or generate a **Translation Memory** from previous translations.

To translate the documents, the **Translation Environment** offers common editor functions and specific functions for translation, such as dictionary lookup and copying translation proposals. Using these functions helps you translate documents more efficiently.

When you translate several documents on the same subject, the **Translation Memory** used will grow, and the number of translation proposals will increase.

Prerequisites

- The document must have been imported.
- To begin translating a document, it must be opened.

Calling sequence

Select:

- The document to be translated from the "Document List" window
- **Open** from the **File** menu

The **Translation Environment** is displayed. It consists of a number of windows. The selected document is shown in the "Translation" window. If the document you selected has already been partially translated, the cursor is positioned on the first untranslated segment and you can continue there.

Options and parameters

For information on how to use the editor functions, see [Working with the OpenTM2 editor](#).

Results

You can choose the sequence in which you translate documents. You can also translate parts of a document at a time and continue later until the documents are completely translated. If needed, you can export partially translated documents and give them to other **OpenTM2** users who can continue the translation.

When you have completed the translation of the segments of a document, **OpenTM2** automatically switches to the postediting mode where you can review the translation and check the spelling. The **Translation Memory** you used contains all segments and their translations and can be used for documents containing similar texts. You can postedit it whenever a translation needs to be modified.

Opening and translating several documents

translatingseveral documents documenttranslating several documents documentopening several documents openingseveral documents You can open several documents and translate them in parallel in the Translation Environment.

If the selected documents have not been analyzed, **OpenTM2** automatically analyzes them based on the properties of the documents and the folder. For more information on the analysis, see [Analyzing a document](#).

Prerequisites

The "Document List" window of the respective folder is active.

Calling sequence

You have two choices:

- You select several documents from the "Document List" window and then **Open** from the **File** menu.
- You open one or more documents while translating using **Open...** from the **File** menu (see "Viewing other documents while translating" on page ...).

Results

In either case, the opened documents are displayed in the "Translation" window together with the respective dictionary and **Translation Memory**. If you used the first choice, the document selected first is the active one. If you used the second choice, the document opened last is the active one.

You can switch to another opened document by either clicking it or by selecting it from the **Translation Environment** option from the **File** menu.

You can close the opened documents one by one.

Viewing the properties of a document

The main properties of a document are:

- **Translation Memory** The **Translation Memory** to be used for the translation of the document.
- **Source language** The language of the original document.
- **Target language** The language of the translated document.
- **Editor** The editor associated with the document.
- **Markup table** The markup table associated to the document. The markup table is used during analysis and translation of the document.

Prerequisites

The document must be in a folder.

Calling sequence

Select:

1. A document from the "Document List" window
2. **Properties Summary...** from the **File** menu

Results

A document in HTML format is displayed containing a summary of the document properties (see [Figure 75](#)).

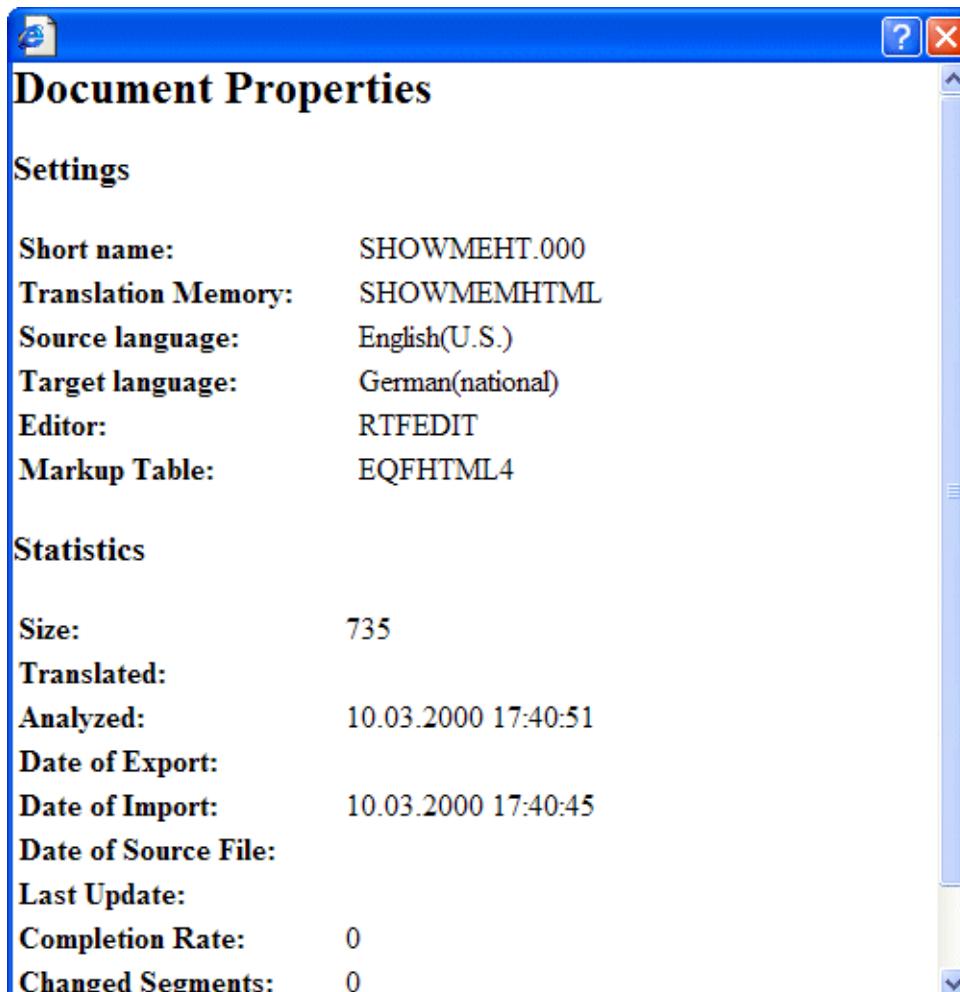


Figure 75. Document Properties document

If you want to view more properties or change the properties of a document, refer to [Changing the properties of a document](#).

Changing the properties of a document

documentchanging properties The properties must be defined individually for each document when you import documents in external format. They can be changed later.

Prerequisites

The document must be in a folder.

Calling sequence

Select:

1. A document from the "Document List" window
2. **Properties...** from the **File** menu

The "Document Properties" window (see [Figure 76](#)) is displayed.

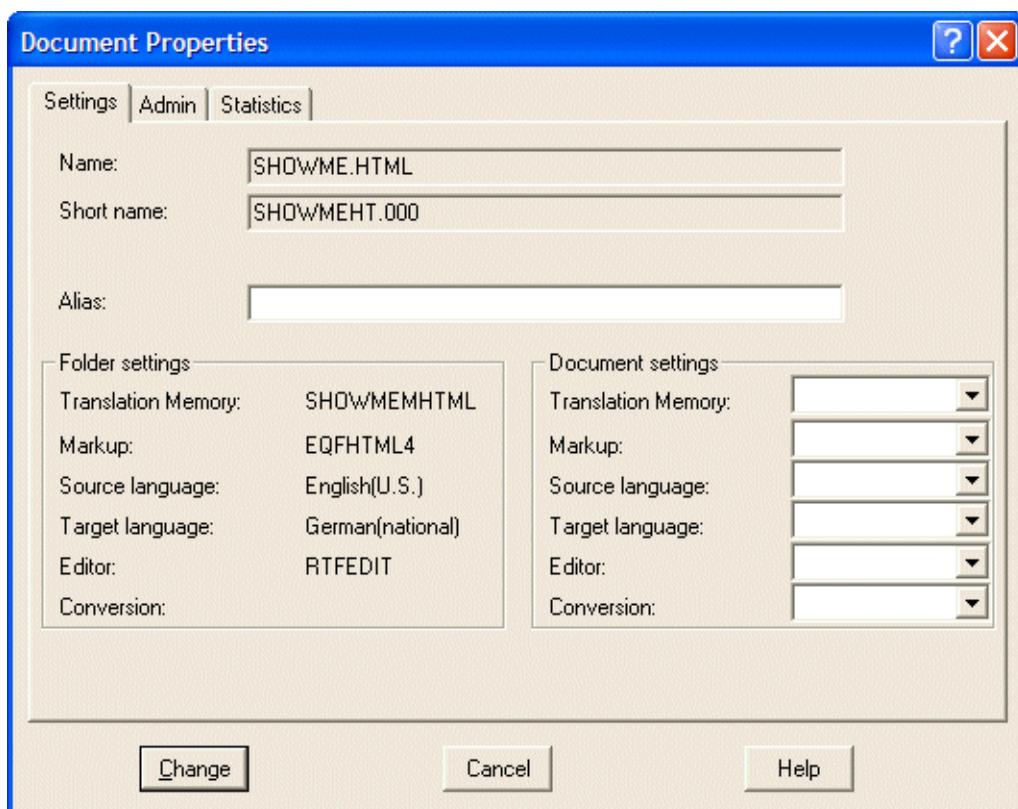


Figure 76. Document Properties window

Options and parameters

On the "Settings" page:

Short name

This field shows the document short name. This name is the internally used file name of the document.

Alias

Either this field contains the alias name that you specified during the import of the document or you can specify an alias name here.

An alias name is useful if you want to get the same exact matches from the **Translation Memory** as for another document that you already translated. You then assign the name of the other document as alias name for this document. Assume, for example, that document `docupdate` is an update of `docnew` and you want to get the same exact matches from the **Translation Memory** that `docnew` received. In this case, you would assign `docnew` as the alias name to `docupdate`.

If an alias name exists, **OpenTM2** uses this name instead of the actual document name to access the correct **Translation Memory**.

The same alias name can be assigned to several documents. An alias name can have up to 256 characters.

Folder settings

This box contains the properties that are specified for the folder in general. They cannot be changed in this window.

Document settings

This box contains the individual settings per document. Initially these boxes are empty.

If the document has the same properties as the folder, you do not have to enter anything, and the document "inherits" the folder properties. Document properties that are different from the folder properties can be selected from the respective list boxes. In the **Translation Memory** field, you can type the name of a new **Translation Memory**.

On the "Admin" page:

Folder administration

Contains part of the information specified on the "Admin" page of the "Folder Properties" window. You cannot change any of the fields.

Document administration

Contains administration information about the document:

Translator's name

If the document is still to be translated, specify or select the name of the person to receive this document next. This can be the actual translator or the vendor, for example. If the document is already translated and on its way back to the project coordinator, leave this field blank.

Translator's e-mail

If the document is still to be translated, specify or select the e-mail address of the person to receive this document next. This can be the actual translator or the vendor, for example. If the document is already translated and on its way back to the project coordinator, leave this field blank.

Shipment number

Contains the shipment number assigned to the folder containing this document. You cannot change this field.

The "Statistics" page contains information about the translation status of the document. The data cannot be changed in this window.

Date of source file

dateof source file source filedate of statisticsdate of source file documentproperties The date and time at which the source file was last modified outside OpenTM2.

Size

The number of bytes the document occupies on your disk.

Translated

The date and time at which translation was completed.

Analyzed

The date and time at which the document was analyzed.

Date of export

The date and time at which the document was last exported.

Date of import

The date and time at which the document was last imported.

Last update

The date and time at which the document was last updated in the **Translation Environment**.

Completion rate

The percentage to which the document is already translated.

Changed segments

The number of segments that were copied during the translation from the **Translation Memory** and then changed.

New segments

The number of segments that were entered entirely new without copying a match from the **Translation Memory**.

Copied segments

The number of segments that were copied from the **Translation Memory** and saved unchanged.

To save the specified properties for this document, click **Change**.

Results

The properties are saved for this document.

Viewing the details of a document

You can display parts of the properties and the current processing status of a document in the "Document List" window. This includes such details as the date of import, date of analysis, the percentage of the document that has been translated. Using this data you can easily show the progress of translation work. You can display these details in the "Document List" window.

The details of a document are:

Name

The name of the document

Date of source file

dateof source file source filedate of statisticsdate of source file documentdetails The date and time at which the source file was last modified outside OpenTM2.

Translated

The date when translation of the document was completed

Analyzed

The date when the document was analyzed

Exported

The date when the document was last exported

Imported

The date when the document was last imported

Updated

The date (and optionally the time) that the document was last updated

Size

The number of bytes of the original document

Complete %

The percentage of the document that has already been translated

Changed

The number of segments that were copied from the **Translation Memory** window during translation and subsequently changed

Shipment

The shipment number of the document

Markup

The markup table used for this document

Translation Memory

The **Translation Memory** used for this document

Source language

The language of the original document

Target language

The language into which the document is translated

Editor

The editor used for this document

Alias

The alias name for the document

Source

The last update date and time of the source document

Name without path

The document name without any path information

Extension

The file extension of the document

If nothing is specified for any of the last four items, **OpenTM2** assumes these properties to be the same as for the folder.

For an imported folder, the details in the "Document List" window are defined by the sender. You can change them if you like. You can toggle between two display modes:

- Displaying the names only
- Displaying selected details

You can change the arrangement of the document details in the "Change Details" window described in [Viewing and changing the details of a list item](#).

Including notes for the translator

OpenTM2 enables you to include notes in the source document to be translated. Thus, you can give the translator hints, tips, or instructions. The author of the document might want to inform the translator about the specific meaning of an expression. Or the project coordinator might want to give the translator the correct translation for a word.

Translator's notes can be inserted at any place in a document. You can specify two levels of notes to emphasize different importance, which is then displayed with different colors, fonts, and indentation in the Translation Environment.

Translator's notes in a source document

OpenTM2 supports translator's notes for document types that allow for annotations or hidden text that do not appear in a printed document. For example, SGML documents, BookMaster^(R) documents, and word processors like Microsoft^(R) Word for Windows^(R) Version 6, and WordPerfect, allow for annotations.

OpenTM2 uses this capability to nest translator's notes inside annotations.

The following example shows the source of a BookMaster^(R) document any how the author has added an annotation and several translator's notes. Note that :*annot*. and :*eannot*. are BookMaster^(R) markup tags, which enclose an ordinary annotation. *trnote1*: and *trnote2*: are **OpenTM2** extensions that start a level 1 and level 2 translator's note. The latter need to be nested inside BookMaster^(R) annotations

```
....text...
:annot. This is an ordinary BookMaster annotation from the author,
not a translator's note. OpenTM2 ignores it. :eannot.
....text... :annot.trnote1: Important: Always translate "Insert" with
"Einfuegen" in order to be consistent throughout the documentation
and the menu items. :eannot. ....text... :annot.trnote2: If time:
check for proper translation of "Manual change". :eannot.
```

If you just want to add translator's notes, you can specify them by adding *trnote1*: or *trnote2*: markups inside annotation markups of a document. Both markups are identical for all supported document types. Upper case notation is supported.

If you work with markup tables in more detail, note that the *trnote1*: and *trnote2*: markups are defined in the markup tables that support an annotation or hiding method.

Translator's notes in the Translation Environment

<TRNOTE> (translator's note) markup A note in a document is referred to by **<TRNOTE>** in the Translation Environment. You can view all translator's notes contained in the currently displayed document in one of the following ways:

translator's notes [ways to view](#)

viewing [translator's notes](#)

- Double-click **<TRNOTE>**
- Select **Commands** from the **Options** menu, select the **TRNote window** entry, and then press **Run**.
- Use the keys that you assigned to this command using the **Keys** dialog.

- Select the **Translator's note** entry from the **File** menu.

You then get the "Translator's Note" window containing all notes available in your current document. When you double-click a note in this window, the input focus is set on the segment next to the note.

<TRNOTE> is treated like any markup tag in OpenTM2:

<TRNOTE> (translator's note) markups hiding shrinking changing name changing color changing font

- You can hide it or display it in shrink mode.

hiding<TRNOTE> (translator's note) markup

- You can change its name in the field on the "Display" page of the "Customize Translation Functions" window (see). This changes the term in the Translation Environment, but has no effect on how translator's notes are marked up in a source document.

changingname of <TRNOTE> (translator's note) markup

- You can change its foreground and background colors. The defaults are black and cyan, respectively. You might want to change them if you work with documents that contain many translator's notes, or notes that are difficult to perceive in context. See "Customizing colors and fonts" on page ... for details.

changingcolor of <TRNOTE> (translator's note) markup

- You can change its font, style, and size (if you work with the RTFEdit editor) to differentiate it from surrounding text. See "Customizing colors and fonts" on page ... for details.

changingfont of <TRNOTE> (translator's note) markup

RTFEdit editor changing font of <TRNOTE> markup

Preview XML documents

The markup table IBMXML is enabled to perform a XML-document preview. In order to preview the XML document, the steps described below should be followed. XML documents require a style sheet (*.XSL) document in order to be properly displayed. Import the XSL document into the folder and analyze it in order to be used by the preview function or select the XSL document from the hard disk drive.

Step by step description to preview a XML document

1. Import and analyze the XML document.
2. Open the XML document in the translation editor.
3. Either click on "**Translate**"- "**Show Translation**" or click on the preview-icon in the icon list.
4. Window ...opens.

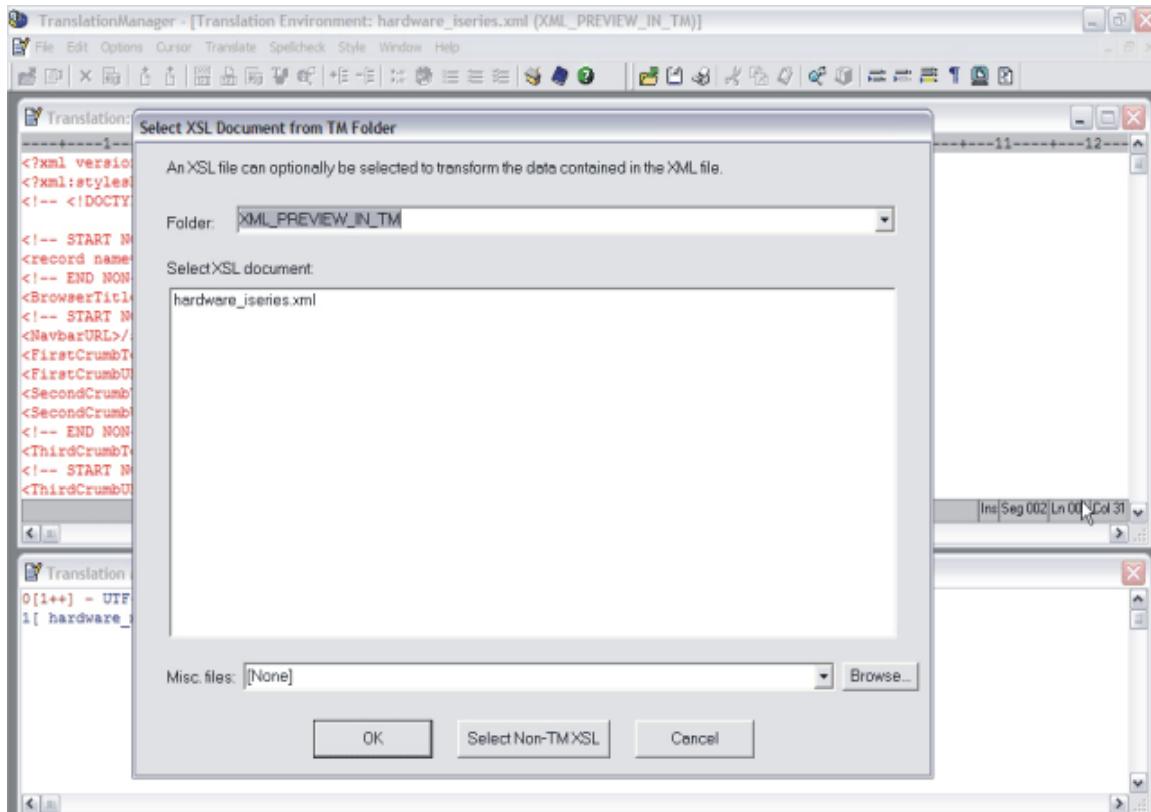


Figure 77. Select XSL document from TM Folder window

5. If the XSL document is part of the folder:

- From select the XSL document which is required to preview the XML document.
- Optionally select a directory where additional files are stored on. This can be graphics, include files etc. Click on button "**Browse...**" and select the appropriate directory in order to fill in the path-information to entry field "**Misc. Files**" or manually type the path to the additional files.

- Click on “OK” and the preview window opens.
6. If the XSL document is not part of the folder:
- Optionally select a directory where additional files are stored on. This can be graphics, include files etc. Click on button “Browse...” and select the appropriate directory in order to fill in the path-information to entry field “**Misc. Files**” or manually type the path to the additional files.
 - Click on button “**Select Non-TM XSL**” in order to select the appropriate XSL-document from the hard disk drive. The preview starts automatically.

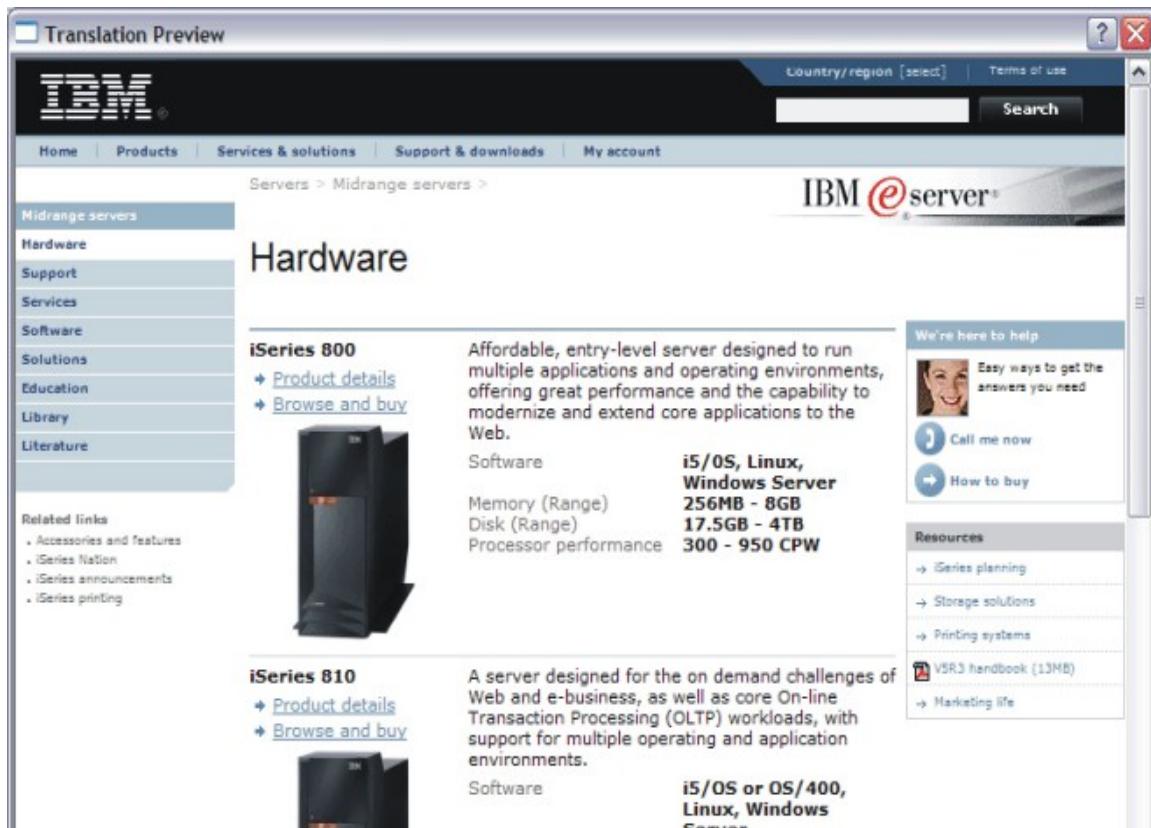


Figure 78. Translation Preview window

Working with folders

Similar to office folders that contain paper files, a OpenTM2 folder contains a number of related documents or document files, respectively. A folder, for example, might contain all documents of a specific translation project, or any other collection of documents that belong together.

Each folder has a number of *properties*, such as the source and target language, or references to the dictionaries and Translation Memory databases to be used for analysis and translation of the documents in the folder. By default, all documents in a folder have the same properties, but you can specify individual properties for each document.

Overview and terminology

You can send all the documents in a folder to another OpenTM2 user.

Folders that are sent or exported contain a number of documents, associated *dictionaries* and **Translation Memory databases**, and optionally an accompanying note to the recipient. To work with a folder that has been exported by another user, you must first *import* it into OpenTM2.

Exporting folders is also useful to make backup copies of your translation material. **controlled folder handling** When creating a folder you can select *Controlled folder handling*. Selecting this option means that you are the project coordinator responsible for this folder. It enables you to specify, and change at any time, all properties and details for the folder, including the translators for the documents to be imported into this folder.

The originator and translator that you specify for the folder also outline the way the folder is taking until it is received by the actual translator, for example, from project coordinator to the vendor, to translator A, to translator B who actually translates the folder contents.

The specifications you make with regard to controlled folder handling appear on various windows and extend your possibilities concerning a folder and its contents. For example, you can export only the documents that are to be translated by a specific translator, a task that was not possible before controlled folder handling was introduced.

Users who are not authorized for controlled folder handling have only restricted rights.

What you can do with folders

OpenTM2 includes a number of processing functions for folders, which you can access by selecting a folder from the "Folder List" window and then selecting a choice from the **File**, **View**, or **Utilities** menu.

Prerequisites

At least one folder must exist, except if you import a folder.

Calling sequence

Select the "Folder List" window from the main window. The "Folder List" window is displayed.

Results

This window displays a list of the available folders. When you select a command from one of the menus, you are taken to another window where you can specify processing options.

Analyzing all the documents in a folder

analyzing all documents in a folder **analyzing the documents it contains** Documents that are to be translated must first be *segmented*. Segmentation is done during *analysis* of a document. You can analyze all the documents in a folder at once.

Prerequisites

- The folder must contain at least one document.
- If you intend using dictionaries to perform analysis, at least one dictionary must exist.
- If you intend using **Translation Memory databases** to perform analysis, at least one **Translation Memory** must exist.

Calling sequence

Select:

1. The folder from the "Folder List" window
2. **Analyze...** from the **File** menu

The "Analyze Documents" window is displayed (see [Figure 44](#))

Options and parameters

Documents to be analyzed

This box lists all the documents in the folder.

For a description of all other options, see [Analyzing documents using Translation Memory databases](#) and [Analyzing documents using dictionaries](#).

When you have selected the desired options, click **Analyze** to begin analysis of the documents.

Results

All documents in the folder are segmented. Depending on the options you selected:

- Terminology lists may be created
- Dictionaries may be extended
- **Translation Memory databases** may be extended
- A file containing untranslated segments may be created
- Segments in the documents may be automatically substituted

You can now begin translating the documents by opening a document, see [Translating a document](#).

Closing a folder

folder closing To start to work with a document, you must first open the folder containing the document. Then the "Document List" window is displayed. Closing a folder means to close the "Document List" window, which takes you back to the "Folder List" window.

Prerequisites

The "Document List" window of the folder must be active.

Calling sequence

Select:

1. The "Document List" window of the folder you want to close
2. **Close** from the system menu of this window

This takes you back to the "Folder List" window.

Results

You can begin working on documents in other folders or any other tasks in **OpenTM2**.

Counting the words in all documents of a folder

foldercounting markup tags in the documents it contains foldercounting words in the documents it contains **OpenTM2** provides a facility for counting the number of words and markup tags in all documents in a folder before and after translation. You can also perform this task for selected documents in a folder (see [Counting the number of words in a document](#)).

Prerequisites

The folder must contain at least one document.

Calling sequence

Select:

1. The folder from the "Folder List" window
2. **Count Words...** from the **Utilities** menu

"Count Words" window (see [Figure 47](#)) is displayed.

Options and parameters

The options and parameters are the same as described in [Figure 47](#).

Click **Count** to begin counting words.

Results

The results are displayed in the "Word Count Results - Original" window or in the "Word Count Results - Translation" window which are explained in [Figure 47](#).

When you close this window, you are returned to the "Folder List" window. If you selected the output option **File**, the results are also saved in a file with the name you choose.

Creating a folder

foldercreating creatingfolders Grouping documents in a folder is recommended whenever:

- A number of documents belong to the same translation project.
- You use the same **Translation Memory databases** for a number of documents.
- You use the same dictionaries for a number of documents.

A folder also makes it easier to export or import a group of documents.

Prerequisites

- Decide for which documents, dictionaries, and **Translation Memory databases** you wish to create a folder.
- At least one language support file must be installed.

Calling sequence

Select:

1. The "Folder List" window in the main window
2. **New...** from the **File** menu

The "New Folder" window (see [Figure 79](#)) is displayed.

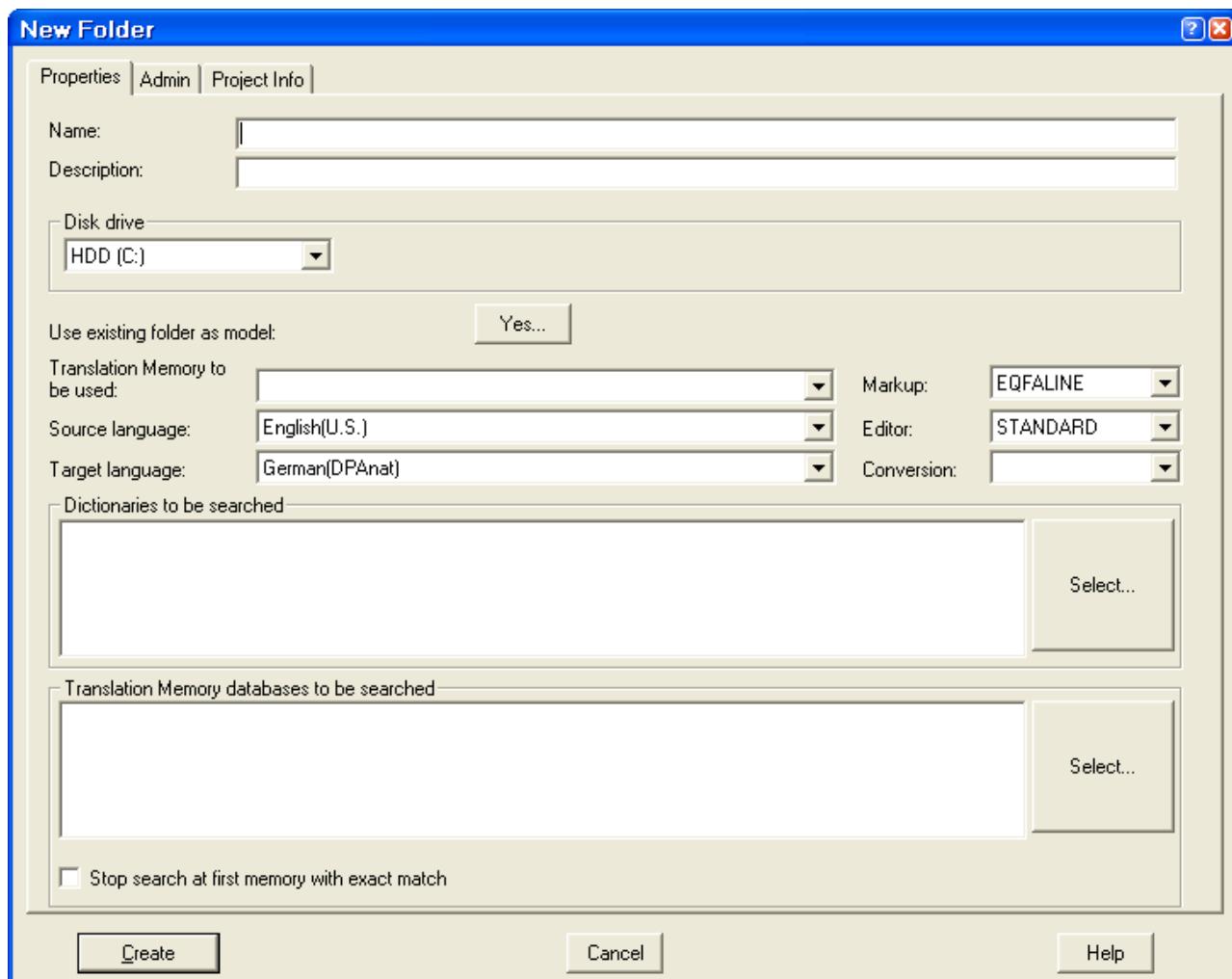


Figure 79. New Folder window
[\[PIC\]](#)Figure shows New Folder window

Options and parameters

The options you specify are used for all documents imported into this folder. If individual documents need other properties, you can specify these when importing the documents. On the "Properties" page:

Name

Type a name of up to 8 alphanumeric characters for the new folder.

Description

Type up to 40 alphanumeric characters that describe the folder.

Disk drive

Select the drive where the folder is to reside.

Use existing folder as model

Click **Yes...** if the properties of the new folder are similar to an existing one. This takes you to the "Use existing Folder as Model" window, where you can select an existing folder as a model for the new folder. Click **Select** to return to the "New Folder" window.

Translation Memory to be used

Select a **Translation Memory** to be used for the documents in the new folder. To create a new **Translation Memory**, type in its name.

Source language

Select the language of the original documents.

Target language

Select the language into which the documents are to be translated.

Markup

Select a markup table name from the available names in the box.

Editor

Select an editor from the available names in the box.

Conversion

Select a conversion from the box that is to be used for all documents in this folder.

Dictionaries to be searched

The selected search dictionaries are shown in the list box. The order in which you select the dictionaries determines the order in which they are searched. You can select up to 10 dictionaries. To select or deselect dictionaries press the **Select...** button which will open the dictionary selection window.

You can also deselect dictionaries by double-clicking the dictionary name.

Translation Memory databases to be searched

The selected search Translation Memory databases are shown in the list box. The order in which you select the Translation Memory databases determines the order in which they are searched. You can select up to four Translation Memory databases. To select or deselect Translation Memory databases press the **Select...** button which will open the Translation Memory selection window.

You can also deselect Translation Memory databases by double-clicking the Translation Memory name.

Stop search at first memory with exact match

Select this option if the search for memory segments should stop at the first memory containing an exact match. This avoids duplicate exact matches from subsequent Translation Memory databases in the list. Without this option OpenTM2 will extract matches from all selected Translation Memory databases.

On the "Admin" page:

Controlled folder handling

Select this option if you are the project coordinator responsible for this folder. Clicking this option enables you to fill in all fields on this page and the "Project Info" page. For more information on this option, refer to [Controlled folder handling](#).

Password

Type a password of up to five alphanumeric characters. This password protects your folder from being changed by unauthorized people on the "Folder Properties" window.

Project coordinator

Specify your name and e-mail address in the appropriate fields.

Originator

Remains empty until this folder has been exported for the first time. After an export it contains the name and the e-mail address of the person who last exported it.

Translator

Specify the name and the e-mail address of the person receiving the folder next. This can be a vendor or the actual translator.

The "Project Info" page enables you to specify all necessary information about the project reflected by this folder. Always specify a shipment number to ensure that you always get a correct Calculating, Preanalysis, or Redundancy Report (see [Creating reports](#)). If you specify the shipment number at a later point of time using the "Folder Properties" window, any documents that the folder then contains need to be exported and imported again to guarantee a correct report.

Click **Create** to create the new folder.

Click **Cancel** to leave this window without creating a new folder.

Results

If you clicked on **Create**, the folder is created, an empty "Document List" window is displayed, and the folder can be filled with documents. If you entered a new **Translation Memory** name as a folder property, you are prompted with the "New **Translation Memory**" window, to specify some more parameters.

Dictionary Selection window

folderselecting creatingdictionary

Options and parameters

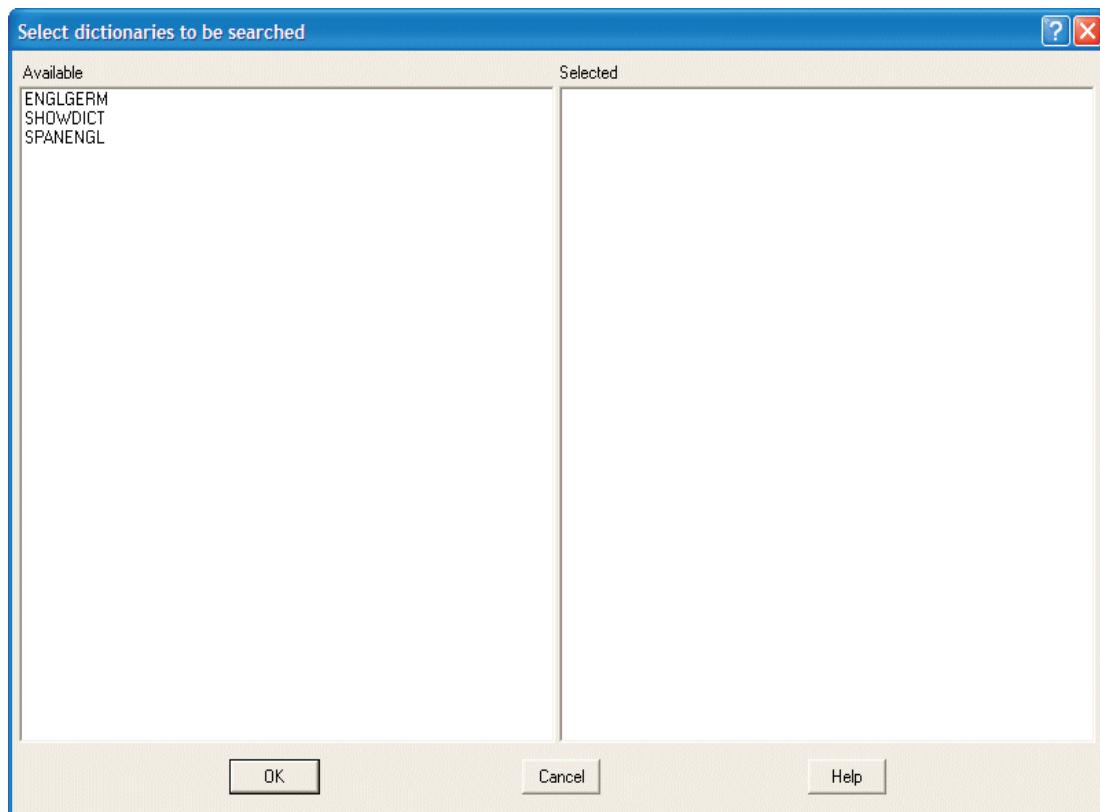


Figure 80. Dictionary Selection window
[PIC]Figure shows Dictionary Selection window

- The **Available** listbox shows the names of the available dictionaries. To add a dictionary to the listbox either double-click the dictionary name or select it using the keyboard and press the "Enter"-key.
- The **Selected** listbox lists the names of the selected dictionaries. To remove a dictionary from the listbox either double-click the dictionary name or select it using the keyboard and press the "Enter"-key.

Press the **OK** button to leave the selection window and return to the previous window or press the **Cancel** button to leave the window and discard any changes made to the list of selected dictionaries.

Translation Memory Selection window

Translation Memory Selection window

folder selecting creating Translation Memory

Options and parameters

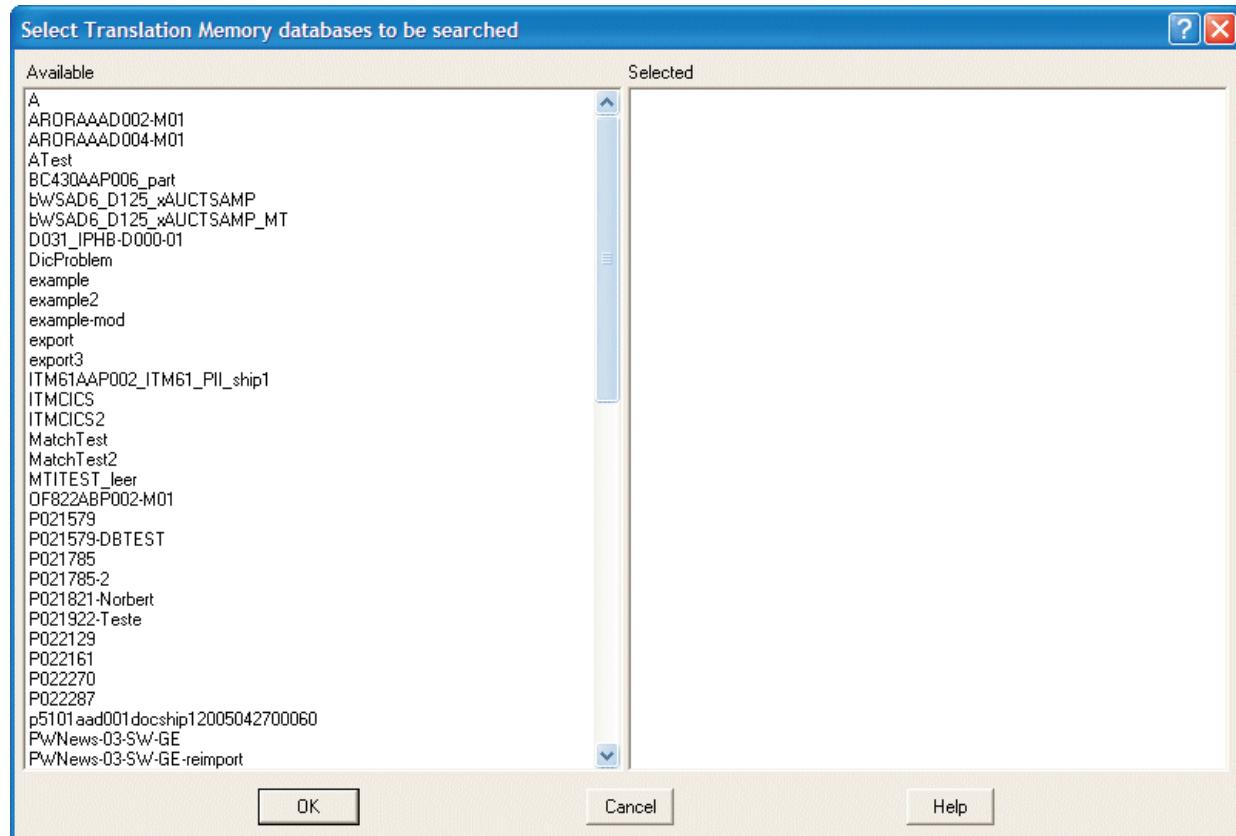


Figure 81. Translation MemorySelection window
[PIC]Figure shows Translation Memory Selection window

- The **Available** listbox shows the names of the available Translation Memory databases. To add a Translation Memory to the listbox either double-click the Translation Memory name or select it using the keyboard and press the "Enter"-key.
- The **Selected** listbox lists the names of the selected Translation Memory databases. To remove a Translation Memory from the listbox either double-click the Translation Memory name or select it using the keyboard and press the "Enter"-key.

Press the **OK** button to leave the selection window and return to the previous window or press the **Cancel** button to leave the window and discard any changes made to the list of selected Translation Memory databases.

Creating a subfolder

When your folder list is in Explorer-like tree view, you can create subfolders to each folder. You can then import documents into the subfolders like into any other folder. You can even create a subfolder for a subfolder.

Prerequisites

Your folder list must be in Explorer-like tree view.

Calling sequence

1. Double-click the folder for which you want to create a subfolder. The contents of this folder is displayed.
2. Click in the area containing the folder contents.
3. Click **New** on the **File** menu.

The "Create a New Subfolder" window (see [Figure 82](#)) is displayed.

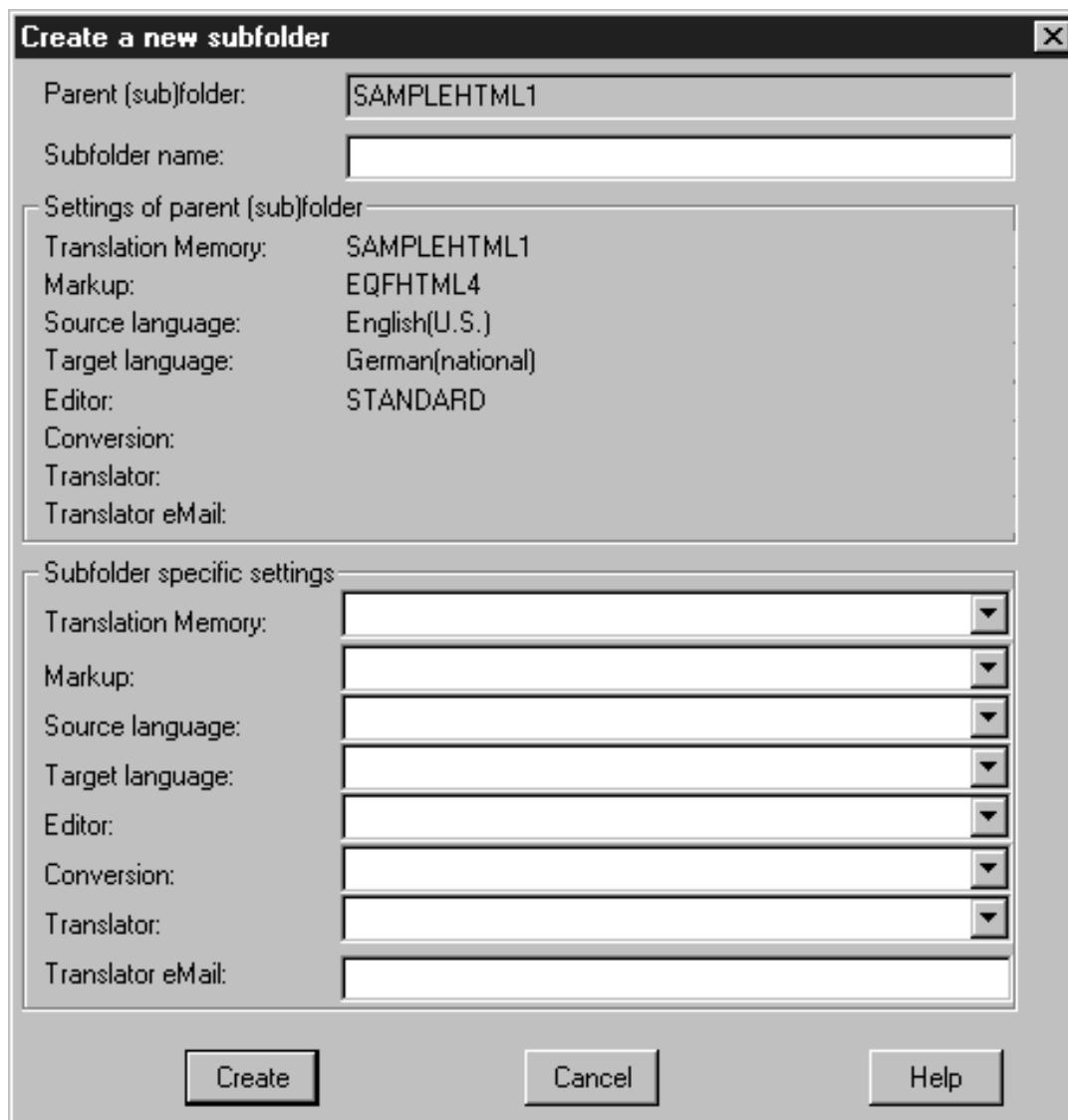


Figure 82. Create a New Subfolder window
[PIC]Figure shows Create a New Subfolder window

Options and parameters

The options you specify are used for all documents imported into this folder. If individual documents need other properties, you can specify these when importing the documents.

Parent (Sub)Folder

Contains the name of the folder or subfolder for which you are creating a subfolder.

Subfolder name

Type up to 40 alphanumeric characters that describe the subfolder.

Settings of parent (sub)folder

Displays the properties of the folder or subfolder for which you are creating a subfolder.

Subfolder-specific settings

If your subfolder is to differ in one or more properties from its parent folder or subfolder, specify them in the appropriate fields.

Click **Create** to create the new subfolder.

Click **Cancel** to leave this window without creating a new subfolder.

Results

If you clicked **Create**, the subfolder is created and added, in the tree view, as a subfolder to its parent folder.

Renaming a folder

folderrenaming renamingfolders **OpenTM2** enables you to give a folder a new name.

Prerequisites

The folder to be deleted must not be open (that is, the "Document List" window for the folder must not be displayed).

Calling sequence

Select:

1. The folder from the "Folder List" window
2. **Rename** from the **File** menu

Options and parameters

Rename to

Type a new name for the folder.

Results

If you select **Rename**, the folder and any references to it are renamed.

Deleting a folder

folderdeleting deletingfolders When you delete a folder, it is no longer listed in the "Folder List" window, and all subfolders and documents in the original and the translated version are also deleted. You might want to delete a folder when you completed working on the documents in the folder, exported it, and no longer require it. Also, if you have backed up the contents of the folder, you can delete the folder to free disk space on your workstation for other purposes.

You can also delete a folder during or after export. For more information, see [Exporting a folder](#) and [Deleting a folder exported to the eqlexport subdirectory](#).

Prerequisites

The folder to be deleted must not be open (that is, the "Document List" window for the folder must not be displayed).

Calling sequence

Select:

1. The folder from the "Folder List" window
2. **Delete** from the **File** menu

Options and parameters

None.

Before deleting the selected folder, **OpenTM2** prompts you to confirm that you want to delete the folder and its contents:

- Select **Yes** to delete the folder.
- Select **No** to leave the folder unchanged.

Results

If you select **Yes**, the folder and the documents in it are deleted. The disk space it occupied is freed. Dictionaries, markup tables, and the **Translation Memory databases** associated with the folder or its documents are not deleted.

Deleting a folder exported to the eqf\export subdirectory

folderdeleting an exported deletingexported folders When you export a folder, **OpenTM2** writes the data to a file in the \eqf\export subdirectory on the selected drive. If you no longer need this exported folder file, you can delete it.

Prerequisites

- The folder must have been exported to eqf\export
- The exported folder must still exist on the drive that you selected when exporting the folder.

Calling sequence

Select:

1. The "Folder List" window in the main window
2. **Delete Exported Folder...** from the **Utilities** menu

The "Delete Exported Folder" window (see [Figure 83](#)) is displayed.

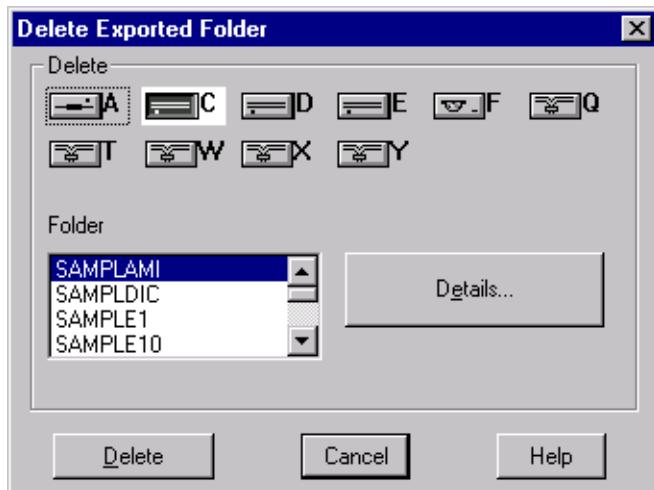


Figure 83. Delete Exported Folder window

Options and parameters

Delete

In this box, select:

1. The drive where the exported folder resides. The **Folder** list box displays all folders previously exported to the selected drive.
2. The folder to be deleted from the **Folders** list box. To see more information about the exported folder, click **Details....** The information is displayed in the "Folder Details" window.

Click **OK** to return to the previous window.

To leave the exported folder unchanged, click **Cancel**.

To delete the exported folder, click **Delete**. For each selected exported folder, a message window prompts you to confirm that you want to delete the exported folder:

- Select **Yes** to delete the exported folder.
- Select **No** to leave the exported folder unchanged.

If there are no more exported folders selected to delete, click **Cancel** to return to the "Folder List" window.

Results

If you confirmed to delete the exported folder in the \eqf\export subdirectory, it is deleted and cannot be reimported.

Exporting a folder

folderexporting exportingfolders The main purpose of exporting a folder is to make documents available to other users or on other systems. Other translation material, such as dictionaries and **Translation Memory databases**, can also be exported, either as part of a folder, or separately. Exporting a folder is the easiest way of distributing translated material from one **OpenTM2** user to another, and it is the recommended way.

You can also include a note for the recipient. This note could, for example, include instructions about which document to translate first or when the translation is due. The recipient can display this note when importing the folder.

When you export a folder to a diskette drive and the folder data volume exceeds the capacity of one diskette, the system writes the data to as many diskettes as necessary.

You can also export the folder as a XLIFF package to allow translation outside of OpenTM2. The export as XLIFF package only exports the translatable parts of the documents together with matching memory proposals. As not all parts of the documents are contained in the XLIFF package it cannot be used to transport or backup a folder. XLIFF packages can only be imported into the folder from which they had been exported.

Prerequisites

The folder to be exported must exist.

Calling sequence

Select:

1. The folder from the "Folder List" window
2. **Export** from the **File** menu

The "Export Folder" window (see [Figure 84](#)) is displayed.

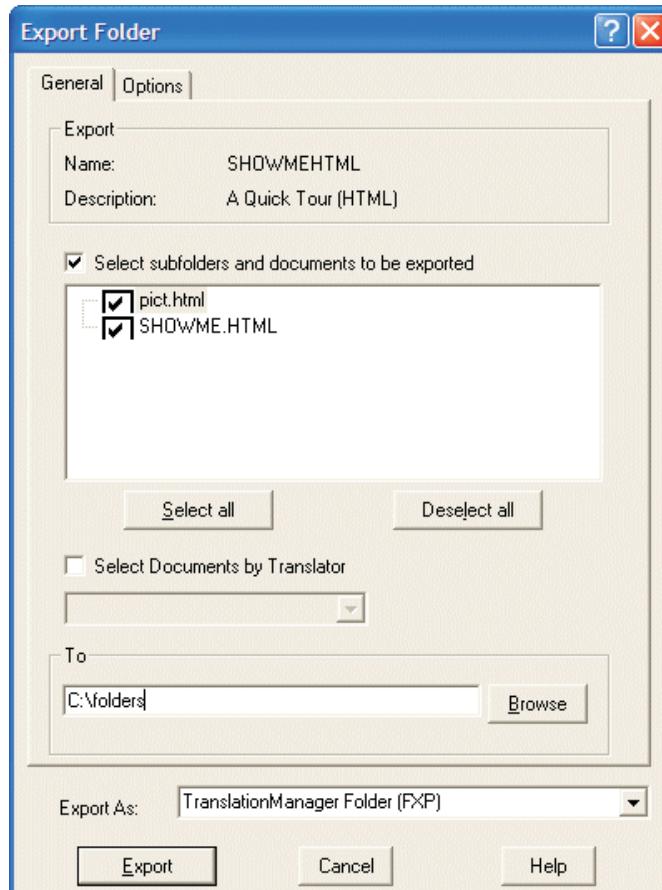


Figure 84. Export Folder window
[PIC]Figure shows Export Folder window

Options and parameters

On the "General" page:

Select subfolders and documents to be exported

To export only selected subfolders and documents of this folder. You can then select these

subfolders and documents from the list box. You can extend or limit your selection by clicking **Select all** or **Deselect all**, respectively.

Select documents by translator

To export only the documents that are to be translated by a specific translator. Select one of the translators listed.

To

Specify the drive and directory to which you want to export your folder. You can also click **Browse** to display the "Browse for Folder" window on which you can select the appropriate drive and directory.

Export as

Export format for the folder. Either select "OpenTM2 Folder (FXP)" for exporting in the standard TranslationManager format or "XLIFF folder" to export the folder as XLIFF package for the translation outside of TOpenTM2.

On the "Options" page:

Word-count data only

To export only the history log file (HISTLOG.DAT) of the folder. For more information on this file, refer to [Overview and terminology](#).

With dictionaries

To export the folder with the dictionaries associated with this folder.

With Translation Memory

To export the folder with the Translation Memory database associated with this folder.

With search Translation Memory databases

To export the folder with the **Translation Memory databases** that are only used to search for matching segments.

With document Translation Memory databases

To export the folder with all the **Translation Memory databases** defined for the folder and its documents.

Export folder as

To export the folder under a different name.

Export using folder short name

To export the folder under a short name. This is useful if you later want to import the folder into a system that recognizes only file names with a length of eight characters plus a 3-character extension.

Export in non-Unicode format

This option exports the folder in a format which is compatible to TP5.5.1 and TP5.4 (non-Unicode versions of TM). Older versions of OpenTM2 can not use folders exported in non-Unicode format due to changes in the file structure. If you use this option the segmented files, contained dictionaries and memories are converted to ASCII and stored in TP5.5.1 format. If you import a folder which is in non-Unicode format you will need to organize the contained memories and dictionaries after the import is completed to be able to use them.

Add note

To add some explanatory text for the recipient of the folder. You type the text in the "Add Note to Folder" window that is displayed when the exporting function is started.

Your note is saved and included in the exported folder.

Delete the folder after export

To delete the folder after it is exportd. This option can only be selected when all documents in the folder are being exported. After the export file is created, the system prompts you to confirm that you want to delete the folder. If you selected the **Select subfolders and documents to be exported** option, the **Select documents by translator** option, or both, you cannot all documents are exported and the folder cannot be deleted.

Export as master folder

To export the folder as master folder. This option can be used to make a backup of a master folder or to move a master folder to another machine. If you export a master folder without specifying this option, the folder will always be imported as a child folder. This option is only active for master

folders.

Originator

Specify your name and e-mail address. Make sure that you filled in these fields before exporting the folder.

Note: Only the option "With Translation Memory" can be selected when exporting the folder as XLIFF package

If you selected the required options, select **Export** to start the export procedure. If you selected the **Add note** option, you are prompted with the "Add Note to Folder" window to type your note for the recipient. To save the note in the exported folder, click **Add** to start the export of the folder.

If you selected a diskette drive and the data volume of the folder exceeds the capacity of a diskette, you are prompted to insert as many diskettes as necessary.

Results

Export as OpenTM2 folder:

- The folder is exported, including all material you selected. If you exported the folder together with documents, any markup tables used by documents are also exported.
- **OpenTM2** stores the exported folder in one file in the specified directory. From this directory it can be reimported. The file has the folder name and the extension **.FTP**
- If you selected the **Delete the folder after export** option, the folder is deleted, including all documents in it. However, the referenced markup tables, dictionaries, and **Translation Memory databases** are not deleted.

Export as XLIFF folder:

- The translatable information of the documents in the folder is exported as XLIFF package. Translation Memory proposals are added to the XLIFF package when the "With Translation Memory" option has been selected. The XLIFF package can be translated outside of OpenTM2 using a XLIFF translation editor. The translated XLIFF package has to be imported into the original OpenTM2 folder before the translated documents can be exported using document export.

Importing a folder

importingfolders folderimporting When translations are prepared centrally before being distributed to individual translators, you are strongly recommended to use folders as the means of distributing this material, because all related translation material (documents, dictionaries, and **Translation Memory databases**) can be included in one folder.

Before you can start to work with material that has been exported, you must import the material in **OpenTM2**.

A folder exported to a diskette drive can span more than one diskette. The system prompts you to insert the diskettes into the diskette drive in the same sequence they were used during export.

Prerequisites

Ensure that there is sufficient free disk space on your system.

Calling sequence

Select:

- The "Folder List" window from the main window
- **Import...** from the **File** menu
-

The "Import Folder" window (see [Figure 85](#)) is displayed.

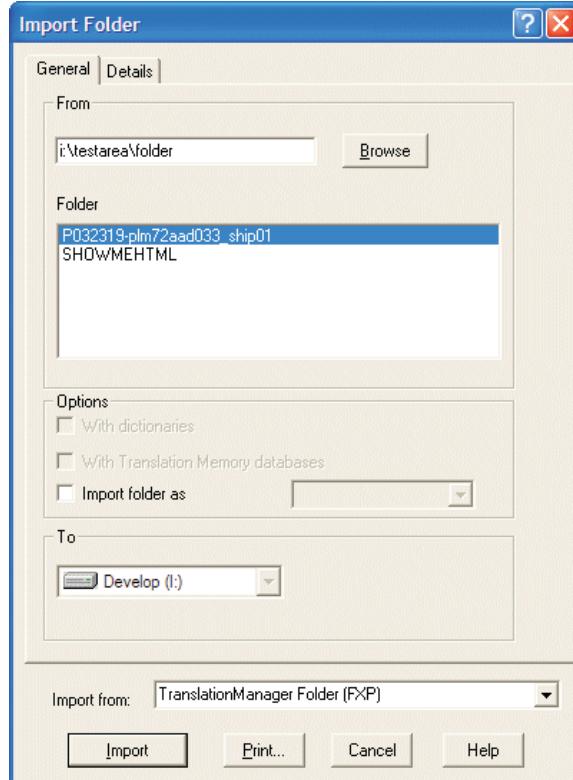


Figure 85. Import Folder window

[PIC]Figure shows Import Folder window

Options and parameters

You can import a folder from any drive that is defined to your installation.

On the "General" page:

From

Select:

- The drive where the folder to be imported resides.
- The folder from the **Folder** box. When you have selected a drive, this box shows the importable folders on this drive.

Options

Select:

With dictionaries

if you wish to import all dictionaries that were previously exported together with the folder. If there are no dictionaries in the folder, this option cannot be selected.

With Translation Memory databases

if you wish to import all **Translation Memory databases** that were previously exported with the folder. If there are no **Translation Memory databases** in the folder, this option cannot be selected.

Import folder as

if you wish to import the folder under a different name.

To

Contains the drive on which **OpenTM2** resides. If you have several **OpenTM2** installations, select the appropriate drive.

Import from

Select "OpenTM2 folder (FXP)" for the import of OpenTM2 folders.

The "Details" page contains information about the selected folder; it cannot be changed on this window.

Name

The name of the folder.

Description

Additional information about the folder.

Date of export

The date and time when the folder was exported.

Translation Memory

The name of the **Translation Memory** referenced in this folder.

Markup

The name of the markup table used.

Export options

This field can contain **Selected documents only** if the folder was exported with this option, otherwise this field is empty.

Documents

If the folder contains documents, their file names are displayed here.

Translation Memory databases

If the folder contains **Translation Memory databases**, their names are displayed here.

Dictionaries

If the folder contains dictionaries, their names are displayed here.

Note

If the folder contains a note, it is displayed here.

To print a copy of the details, click **Print**. To begin importing the folder, click **Import**. If you import the folder from a diskette drive and the exported folder was written to more than one diskette, you are prompted to insert the diskettes in the sequence in which they were inserted during export.

Results

The folder is imported. It is added in the "Folder List" window, and you can start using its contents or the imported dictionaries for document analysis.

If the folder contains a dictionary with the same name as an existing one, you are asked whether the dictionary is to be ignored, or to be imported and automatically merged with the existing one. If the folder contains a **Translation Memory** with the same name as an existing one, you are asked whether the **Translation Memory** is to be ignored, or to be imported and automatically merged with the existing one. An imported folder also contains the markup tables of the folder and document properties. During import, markup tables in your system are overwritten if they have the same name and the markup table in the folder is a more recent one.

If you import a folder that already exists but the content of one or more of its documents differs from the existing ones, the "Set Document Processing Options" window is displayed. You can then decide for each document whether it is to be imported. When you open the imported folder, the "Document List" window displays the details arrangement defined by the sender.

Importing a XLIFF folder

importingfolders (XLIFF) folderimporting Use the folder import in XLIFF format to import XLIFF packages translated outside of OpenTM2.

Prerequisites

The original XLIFF package has to be exported from a OpenTM2 folder. The original OpenTM2 folder has to exist.

Calling sequence

Select:

1. The "Folder List" window from the main window
2. **Import...** from the **File** menu

The "Import Folder" window (see [Figure 86](#)) is displayed.

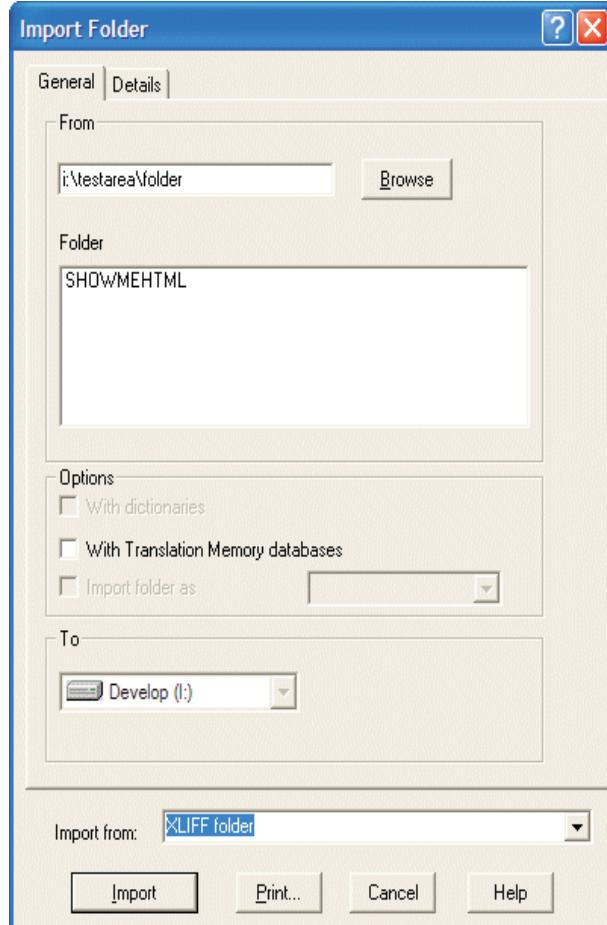


Figure 86. Import Folder window

[PIC]Figure shows Import Folder window

Options and parameters

You can import a folder from any drive that is defined to your installation.

On the "General" page:

From

Select:

- The drive where the XLIFF package to be imported resides.
- The folder from the **Folder** box. When you have selected a drive, this box shows the importable folders on this drive. The folder box will only list XLIFF packages which have been created by OpenTM2. The box shows the folder names and not the names of the XLIFF package files.

Options

Select:

With Translation Memory databases

if you wish to import the translated segments contained in the XLIFF package to the Translation Memory databases of the folder or document.

To

Contains the drive on which **OpenTM2** resides. If you have several **OpenTM2** installations, select the appropriate drive.

Import from

Select "OpenTM2 folder (FXP)" for the import of OpenTM2 folders.

The "Details" page is empty for XLIFF folder imports.

Results

The translated segments contained in the XLIFF package replace the corresponding segments of the documents. The document completion ratio is updated accordingly. If the "With TranslationMemory databases" option has been selected, the translated segments from the XLIFF package are also added to the TranslationMemory of the folder or document.

Importing a folder into controlled folder



Figure 87. Controlled Folder Password window
[PIC]Figure shows Controlled Folder Password window

This window is shown whenever a folder is imported into an existing controlled folder and the folder being imported is not a child of the controlled folder.

Options and parameters

Password

Enter the password of the controlled folder into which the folder should be imported.

OK

Checks the given password and checks it against the password of the controlled folder. If the password is correct, the folder import starts. If the password is not correct, the folder import window is shown.

Cancel

Closes the "Controlled Folder Password" window and returns to the folder import window.

Help

Gives information to help you to interact with this window.

Opening a folder

folderopening openingfolders You open a folder to see which documents it contains. You can then select a document to work with from the "Document List" window.

Prerequisites

The folder must exist.

Calling sequence

Select:

1. The folder from the "Folder List" window
2. **Open** from the **File** menu

Alternatively, double-click the folder to open it. The "Document List" window is displayed.

Results

A list of documents in the selected folder is displayed. From this list, you can start document-related functions by selecting one or more documents and an action from one of the menus. For further information see [What you can do with documents](#).

Searching and replacing text across several documents in a folder

foldersearching and replacing text in its documents OpenTM2 provides a Global Find and Replace function, which works across several folders or across all or some documents of a folder.

This function might be helpful if you are looking for a particular piece of text whose source document you are not sure of. Also, you may need to change a particular wording, but do not want to type the new text manually at each occurrence. OpenTM2 can search through a series of documents to locate the required text and change it as specified.

Any change is automatically reflected in the associated Translation Memory to be available for future translations. Note that OpenTM2 searches and changes only pieces of text that can be

translated. Markup tags cannot be changed. Wildcard search as well as AND / OR combinations of the search terms are possible. For information on how to search and replace text, refer to [Searching and replacing text in documents](#).

Viewing the properties of a folder

folderdisplaying properties folderproperties The properties of a folder are:

Folder name

The name of the folder.

Document format

The format of the documents contained in the folder.

Source language

The language of the original documents in the folder. A different language can be specified for individual documents in the folder.

Target language

The language of the translated documents in the folder. A different language can be specified for individual documents in the folder.

Description

The description of the folder. The description can be up to 40 alphanumeric characters long, including blanks. This description is for your use and is not used by **OpenTM2**.

Editor

The text editor used (default value STANDARD).

Memory

The **Translation Memory** associated with the folder. This **Translation Memory** is searched during segment matches and is updated when documents in this folder are being translated.

Dictionary

The dictionaries associated with the folder. These dictionaries are used in conjunction with the selected exclusion lists during analysis. They are also used for looking up a term. The order in which you select the dictionaries determines the search order associated with the folder.

Conversion

The conversion to be used for all documents in this folder.

Translation Memory databases to be searched

The **Translation Memory databases** to be searched for segment matches in addition to the **Translation Memory** to be used and updated during translation. These additional databases are not updated during translation.

Prerequisites

The folder must exist.

Calling sequence

Select:

1. A folder from the "Folder List" window
2. **Properties Summary...** from the **File** menu

Results

A document in HTML format is displayed containing a summary of the folder properties (see [Figure 1](#)).

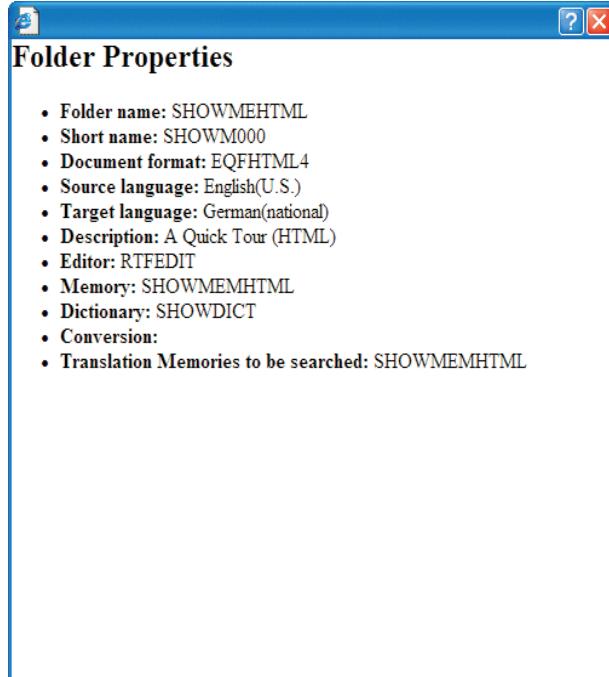


Figure 88. Folder Properties document
[PIC]Figure shows Folder Properties document

If you want to view more properties or change the properties of a folder, refer to [Changing the properties of a folder](#).

Changing the properties of a folder

folderchanging properties folderproperties When you create a folder, you must define its properties before you can work with the folder. Some of them can be changed later.

The properties defined for each folder apply to all documents in the folder unless individual document properties have been set. It is sometimes not advisable to change the properties of a folder. In particular, you should not change the **Translation Memory** associated with the folder when you have started translating a document. You can change the associated dictionaries, if necessary.

Prerequisites

- The folder must exist.
- If you want to change the properties of the folder, it must not be open.

Calling sequence

Select:

1. The folder from the "Folder List" window
2. **Properties** from the **File** menu

The "Folder Properties" window (see [Figure 89](#)) is displayed.

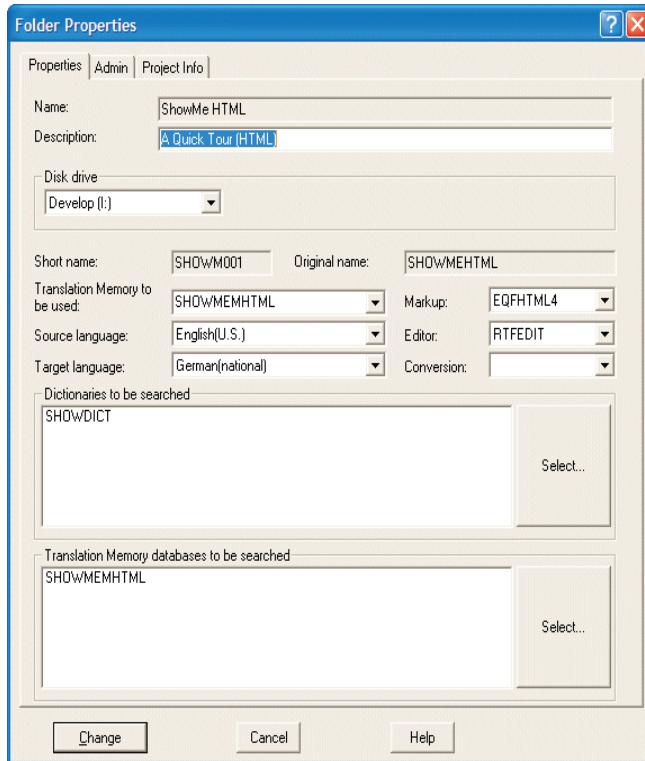


Figure 89. Folder Properties window
[\[PIC\]Figure shows Folder Properties window](#)

Options and parameters

On the "Properties" page, the selected properties are shown in the respective fields. The **Name** field and the **Short name** field cannot be changed for an existing folder.

The following fields contain the currently assigned names. They can be changed by selecting another name from the respective list box.

- **Translation Memory**
- Markup
- Editor
- Source language
- Target language

For the **Translation Memory** to be used, you can also type a new name in the respective field.

The dictionaries to be used in the folder are selected as follows:

- The **Dictionaries to be searched** list box shows the selected dictionaries.
- To select or deselect dictionaries press the **Select...** button, which will open the dictionary selection window.
- Dictionaries can also be deselected by double-clicking the dictionary name.

The **Translation Memory databases** to be searched in the folder are selected as follows:

- The **Translation Memory databases** list box shows the selected Translation Memory databases.
- To select or deselect Translation Memory databases press the **Selected** button, which will open the Translation Memory selection dialog.
- Translation Memory databases can also be deselected by double-clicking the Translation Memory name.

If the folder is protected by a password, you can only change the fields on the "Admin" page if you know the password.

Controlled folder handling

If it is a folder for which controlled folder handling was selected during its creation, this check box is selected. For more information on this option, refer to [Controlled folder handling](#).

Password

If the folder is protected by a password, type the password assigned to this folder.

Project coordinator

Contains the name and e-mail of the project coordinator. You can change the fields if you are the project coordinator or allowed to perform the tasks of the project coordinator.

Originator

If this folder has not been exported yet, the **Name** and the **e-mail** fields are empty. If this folder was exported before, the fields contain the name and e-mail address of the person who last exported it.

Translator

Select or specify name and e-mail address of the person to receive the folder next.

The "Project Info" page contains information about the project. You can change all fields if you are the project coordinator. If the folder is protected by a password, you can only change them if you specified the correct password on the "Admin" page.

If you change the shipment number or specify one for this folder for the first time, export and import again any documents that this folder already contains. This is necessary to get a correct Calculating, Preanalysis, or Redundancy Report (see [Creating reports](#)).

The "Admin" page contains also two buttons for the maintenance of the folder history log. The folder history log contains all information concerning the folder history and is vital for the creation of calculation reports.

- The "Delete history log" button deletes the complete history log. No calculation reports can be performed for this folder anymore. Use this button only when an existing folder is re-used for another translation job or for another shipment and the counting information of the previous shipment is not required anymore.
- The "Clean history log" button tries to reduce the size of the history log by removing superfluous entries from the history log.

To save the changes of the properties, click **Change**.

To leave the "Folder Properties" window without changing the properties, for example, if you only wanted to look at the folder properties, click **Cancel**.

Results

If you select **Change**, the folder properties are changed according to your specifications. Otherwise, they remain unchanged. You are returned to the previous window.

If you entered the name of new **Translation Memory**, the **New Translation Memory** is displayed to specify the characteristics of this **Translation Memory**.

If you changed a markup table name, it is strongly recommended that you analyze the documents in the folder again. For more information how to select a **Dictionary** or a **Translation Memory**, refer to

Viewing the details of a folder

In the "Folder List" window, you can display additional information about a folder. You can define which details you want to be displayed and in which order.

The displayable folder details are as follows:

Name

folderdetails The name of the folder.

Description

The description of the folder.

Updated

The date when the folder properties were last changed.

Markup

The markup table defined in the folder's properties. This markup table is used for all documents in the folder unless a different markup table is specified in the properties of a document.

Translation Memory

The **Translation Memory** associated with the folder.

Drive

The disk drive where the folder resides.

Dictionary

The first dictionary in the list of dictionaries associated with the folder.

Editor

The text editor used (default value STANDARD).

Source language

The language of the original documents in the folder. A different language can be specified for individual documents in the folder.

Target language

The language of the translated documents in the folder. A different language can be specified for individual documents in the folder.

Search Translation Memory

The **Translation Memory databases** to be searched for segment matches.

You can toggle between two display modes:

- Display the names only
- Display selected details

You can change the arrangement of the folder details in the "Change Details" window described in [Viewing and changing the details of a list item](#).

Working with Translation Memory databases

A **Translation Memory** is a database that contains existing translations. You use a **Translation Memory** to find and reuse these translations. [Translation Memorydefinition of](#)

During analysis **OpenTM2** splits a document into translatable segments, which usually correspond to sentences.

A **Translation Memory** contains pairs of such segments that consist of an original (source) segment and a translated (target) segment, which together make up a **Translation Memory** segment. A **Translation Memory** can contain source sentences in one *source language*, and translated sentences in several *target languages*.

Overview and terminology

Translation Memory databases are used during *translation* and *analysis* of documents.

- During translation in the **Translation Environment**, **OpenTM2** displays matching sentences as *translation proposals* for the current sentence. You can then choose the translation proposal that represents the best translation of the sentence, and copy it into the document you currently translate.

The type of match is always evaluated by comparing a sentence in the source document with source sentences in the **Translation Memory**.

OpenTM2 proposes both exact matches and fuzzy matches. [Translation Memory](#) [matchesexact](#) [matchsexact](#) Exact matches are found if two source segments are completely identical. Fuzzy matches are found if two source segments are almost identical. [Translation](#) [Memory](#) [matchesfuzzy](#) [matchsfuzzy](#)

All types of fuzzy matches are indicated by a character enclosed in brackets in front of the proposal. For example, [f] indicates a fuzzy match.

If two source sentences differ by only a digit or a number, **OpenTM2** proposes this as a fuzzy match, with the number already replaced with the new value. Such a *fuzzy replacement* match is indicated by [r]. The various types of matches are described in detail in [Translation Memory matches](#).

Whenever you translate a segment, **OpenTM2** automatically adds your translation to the **Translation Memory** associated with the current document. The **Translation Memory** is therefore being continuously updated—the translation of one sentence may even be offered as a translation proposal for the next matching segment.

- During document analysis, you can choose to have **OpenTM2** automatically replace sentences with matches that already exist in the **Translation Memory**. This substitution occurs only for exact matches.
- During document analysis, **OpenTM2** can also extract segments with an exact or a fuzzy match from existing **Translation Memory databases** and copy these segments into a new one. In this way, you can "preload" a document-related **Translation Memory**.

When the same original sentence appears in several places in a document, **OpenTM2** also allows you to store several different translations of the sentence. In this way, you can translate a segment in different ways according to the context.

Translation Memory databases provide an efficient method of translating whenever you deal with similar or repetitive texts (for example revised editions of manuals).

During translation, you can select *automatic substitution*. This option automatically copies existing translated segments into the translation document for as long as it finds exact matches for the current source segments. Automatic copying stops when no exact match, or more than one exact match, has been found for a source segment. Now you must continue translating the segment manually. Afterwards automatic copying can be restarted.

Translation Memory databases can be even combined with one another. Use the *merge* option to merge the contents of one **Translation Memory** with another.

Translation Memory databases can have two formats:

- Internal, called **OpenTM2** format. The internal format is used to work with **Translation Memory databases** within **OpenTM2**.
- External. The external format is an SGML-based data exchange format that allows the data within **Translation Memory databases** to be processed outside of **OpenTM2**. The data in an external **Translation Memory** is interspersed between SGML tags that describe the structure of the **Translation Memory** as a whole and the data elements of each segment. This format allows you to edit a **Translation Memory** with a text editor, for example to change or delete segments in a **Translation Memory**

If you have translated documents that were translated outside **OpenTM2**, you can generate an *Initial Translation Memory* (*ITM*), this means you can fill a **Translation Memory** with existing translation segments. When you start the translation of a document that is similar to an existing one, you can benefit from your previous translations that are now available in the Initial Translation Memory.

An exported **Translation Memory** in SGML format is also helpful for reusing existing translations. If you have previously saved files or databases in another format, you can access them with **OpenTM2** by using a program to convert them to the format of an external **Translation Memory**.

Translation Memory matches

During translation in the **Translation Environment**, **OpenTM2** searches for matching segments, this means it checks whether the **Translation Memory** contains a previous translation of the current segment.

The system differentiates mainly between an *exact match*, which is completely identical, and a *fuzzy match*, which is an almost identical match where the fuzziness can have various degrees.

Translation Memory matchestypes of matchestypes of

When **OpenTM2** displays a matching segment in the "Translation" window of the **Translation Environment**, the type of fuzzy match is indicated by a preceding flag in brackets, for example [f]. Exact matches do not have such a flag.

An exact match is found when the **Translation Memory** contains a segment identical to the current segment in the document to be translated and when the segment translation was accepted by a human translator during translation. The segments must be identical with respect to text and punctuation but their tagging can differ.

If an identical source segment is placed in the **Translation Memory** by the EQFITM command or was imported as machine-generated translation, it is not regarded as an exact match. It is flagged by a preceding [m].

If you choose a **Translation Memory** previously used for a BookMaster^(R) document, for the translation of machine-readable information, all trailing line feeds and blanks are removed when an exact match is found. This ensures that the target sentence has as many trailing line feeds as the source segment.

By default, a fuzzy match is recognized if the two segments overlap more than 33%. You can customize this threshold for segment lengths, as described in [#unresolvedid/viewsysprop](#). **fuzzy match overlap ratiodefault**

OpenTM2 can also show combinations of these types of matches. The following table shows the categories of matches and the flagging displayed in the "Translation Memory" window.

Translation Memory matchescategories matchescategories	Flag
Type of match	
Exact	(not flagged)
Matches with minor deviations:	
- Machine	[m]
- Replacement	[r]
- Machine and replacement	[mr]
Fuzzy matches:	

Translation Memory matches categories	Flag
Type of match	
- Fuzzy	[f]
- Fuzzy and replacement	[rf]
- Fuzzy and machine	[mf]
- Fuzzy, machine, and replacement	[mrf]

What you can do with Translation Memory databases

Most of the tasks concerning **Translation Memory databases** begin from the "Translation Memory List" window.

Calling sequence

Select:

1. "Translation Memory List" window from the main window
2. A **Translation Memory** from the window
3. One of the enabled commands from the **File** or **View** menu

This usually takes you to another window, where you can specify the necessary options and parameters and where you can start the requested procedure.

Creating a Translation Memory

creating Translation Memory databases Before you can use a **Translation Memory**, it must be initialized. To initialize a **Translation Memory**, you need to specify values for a number of parameters.

The procedure described here creates an empty **Translation Memory**. This can subsequently be filled by translating in the **Translation Environment**, by copying segments from other **Translation Memory databases** during analysis, or by importing external **Translation Memory databases**.

For a description of the other methods of creating and filling a **Translation Memory**, see:

- ["Creating an external Translation Memory"](#)
- ["Merging Translation Memory databases"](#)

Prerequisites

At least one language-support file must be installed.

Calling sequence

Select:

1. The "Translation Memory List" window in the main window
2. **New...** from the **File** menu

The "New Translation Memory" window (see [Figure 90](#)) is displayed.

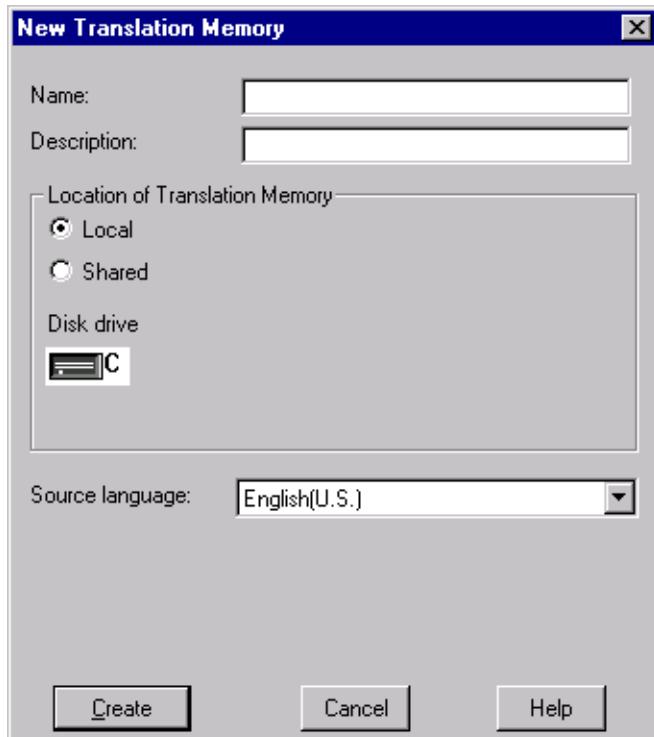


Figure 90. New Translation Memory window

Options and parameters

Name

Type a name for the new **Translation Memory**.

Description

Type a description of the new **Translation Memory**. This description can be up to 40 alphanumeric characters long. It is for your personal use and is not used by **OpenTM2**.

Location of Translation Memory

Specify where you want the new **Translation Memory** to reside. You can keep it on a local disk drive (only you can use it) or on a shared disk drive (several translators can work with it simultaneously). See [#unique_131](#) to learn about the benefits of Translation Memory databases residing on a shared disk drive. **Local** is the default selection. Note that a LAN drive is also considered as local drive.

Disk drive

Shows local or shared disk drives. Select one of the available disk drives as the drive on which you want the new **Translation Memory** to reside. Remember that a **Translation Memory** grows in size over time, so ensure that the drive you select has sufficient free space.

Source language

Select the source language from the list of languages displayed.

Click **Create** to create the new **Translation Memory**.

Results

The **Translation Memory** is created, but is still empty. You can now begin to fill it by translating text or by actions such as import, merge, or analysis.

Creating an external Translation Memory

An external **Translation Memory** is a file that contains segment pairs from previous translations and SGML tags. The SGML tags are used to distinguish between *control information* and the original and translated segments.

SGML format is used as the **OpenTM2** data exchange format for **Translation Memory databases**.

If you have already a translation database in a format of your own and want to use the information in it with **OpenTM2**, you must first convert it into a file in SGML format, for example, by a conversion program that understands your database format and the SGML format. This file can then be imported into **OpenTM2**.

Besides generating an *Initial Translation Memory* (*ITM*), this is another method of filling a

Translation Memory before beginning a translation.

When you export a **Translation Memory**, OpenTM2 creates a file in SGML format. During import, you can also merge an external **Translation Memory** with an internal **Translation Memory**.

Format of an external Translation Memory

A **Translation Memory** in SGML-based format includes, in addition to the original and translation segments, SGML tags that describe the data structure of each segment.

You can also use this format to create an external **Translation Memory** containing your own translation database data.

An external **Translation Memory** is an ASCII coded file. It starts with <NTMemoryDb> and ends with </NTMemoryDb>.

The starting tag is followed by a description of the **Translation Memory**, enclosed between <Description> and </Description> tags.

The description is followed by the segment pairs. The number of segment pairs is not limited. The individual segment pairs are tagged in the following format:

Example

```
<Segment>nnnnnnnnnnn ↵
<Control> ↵
... ↵
</Control> ↵
<Source>...</Source> ↵
<Target>...</Target> ↵
</Segment> ↵
```

The symbol ↵ stands for the combination of the *carriage return* and the *line feed* characters. This character combination is usually generated when you press Enter to return you to the beginning of the next line.

The following describes the SGML structure sequentially.

1. A segment begins with the <Segment> tag.
2. The segment identifier nnnnnnnnn must immediately follow the <Segment> tag and must be 10 characters long. This identifier must be unique within the **Translation Memory**.
3. . ↵ must follow the identifier.
4. The next tag is <Control>, and is followed by ↵.
5. The next line is a sequence of numbers representing the *segment control information*, which is described in “**Control information of a Translation Memory segment**”.
6. The control information must be terminated by the </Control> tag and be followed by ↵.
7. The next tag is <Source>, followed by the source segment. It is ended by </Source>, followed by ↵. If ↵ appears in the source segment, it is considered to be part of the source.
8. The next tag is <Target>, followed by the target segment. It is ended by </Target>, followed by a ↵. If ↵ appears in the target segment, it is considered to be part of the target.
9. The last tag of a segment is </Segment>, followed by ↵.

[The Translation Memory format](#) shows a small part of an external **Translation Memory**.

The Translation Memory format

The following shows part of an SGML-based **Translation Memory** containing two segments. The separator in the segment control information is indicated by the symbol •. The combination of the *carriage return* and the *line feed* characters (↵) is not shown in this example.

```

<NTMemoryDb>
<Description>
This is a Translation Memory sample
</Description>
:
<Segment>0000000001
<Control>
000015•0•0000000668798940•English(U.S.)•German(national)
••EQFBOOK•DEMO.SCR
</Control>
<Source>The amount of data that you wish to process
</Source>
<Target>Die Datenmenge, die verarbeitet werden soll
</Target>
</Segment>
:
<Segment>0000000043
<Control>
000003•0•0000000668798783•English(U.S.)•German(national)
•EVS•EQFBOOK•DEMO.SCR
</Control>
<Source>The IBM 9370 system includes a processor, I/O devices and the
appropriate
software to fulfill your data processing requirements. </Source>
<Target>Das System IBM 9370 umfaßt den Prozessor, E/A Geräte sowie die
entsprechende Software, um den Ansprüchen bei der
Informationsverarbeitung
gerecht zu werden. </Target>
</Segment>
:
</NTMemoryDb>

```

SGML tags for external Translation Memory databases

Start tag, end tag	Description
<NTMemoryDb>, </NTMemoryDb>	Start and end of a Translation Memory
<Segment>, </Segment>	Start and end of a Translation Memory segment
<Description>, </Description>	Start and end of a Translation Memory description
<Control>, </Control>	Start and end of the segment control information
<Source>, </Source>	Start and end of a source sentence
<Target>, </Target>	Start and end of a target sentence

Control information of a Translation Memory segment

The control information for each segment consists of several pieces of information, each of a specific length. The maximum length of each data element is contained in square brackets. The data elements must be separated by a character with the ASCII code 21 ('X'15'), represented by the symbol §.

The control information has the following layout:

Segment number [6]

The segment number forms a number (with leading zeros if required) representing the position of the segment within the document file from which it originates. If you do not know this number, specify a number that is unique within this **Translation Memory**. The segment number is used by the **Translation Memory** to rank segments with an identical source.

Translation indicator [1]

This character can be 0 or 1:

- 0 Indicates that the translation is a manual translation.
- 1 Indicates that the translation of this segment originates from a machine translation system and has not been post-edited by a translator. If more than one exact translation match is available in the **Translation Memory**, **OpenTM2** ranks the human translation higher than the machine translation.

Time stamp [16]

In exported **Translation Memory databases**, this data element contains a time stamp in internal format. In **Translation Memory databases** created outside of **OpenTM2**, use a value of 0.

Source language [20]

Any language from the list of supported source languages.

Target language [20]

Any language from the list of supported target languages.

Author [15]

Contains the name of the translator as the author of the translation of this original segment. It is the only field that is optional.

Markup table [8]

Any of the markup tables available.

File name [12]

Contains the name of the source file from which this segment originates. If you do not know the file name, put any other identification in this field. The **Translation Memory** uses the file name to rank segments originating from the same source. This means if a document with the name XYZ is translated and several exact translation matches are found for a specific segment, then the exact translation with the file name XYZ (if available) is ranked higher. The file name must be immediately followed by **◀—**.

Renaming a Translation Memory

Translation Memory renaming renaming **Translation Memory databases** OpenTM2 enables you to give a **Translation Memory** a new name.

Prerequisites

The **Translation Memory** must exist.

Calling sequence

Select:

1. The **Translation Memory** from the "Translation Memory List" window
2. **Rename** from the **File** menu

Options and parameters

Rename to

Type a new name for the **Translation Memory**.

Adjust all references automatically

If you select this option, all references to the renamed **Translation Memory** are also changed.

Results

If you select **Rename**, the **Translation Memory** and, optionally, any references to it are renamed.

Deleting a Translation Memory

Translation Memory deleting deleting **Translation Memory databases** If you no longer need a **Translation Memory**, for example, if you have specified incorrect parameters or it has been filled with the wrong data, you can delete it.

For learning how to delete a shared Translation Memory, see [Deleting a shared Translation Memory](#).

Prerequisites

The **Translation Memory** must exist.

Calling sequence

Select:

1. The **Translation Memory** to be deleted
2. **Delete** from the **File** menu

Before OpenTM2 deletes the requested **Translation Memory**, a message window is displayed asking you to confirm that you want to delete the **Translation Memory**.

Options and parameters

- If you select **No**, the **Translation Memory** is not deleted.
- If you select **Yes**, the **Translation Memory** is deleted.

Results

If you select **Yes**, the **Translation Memory** is deleted, and the disk space is freed. Otherwise, the **Translation Memory** remains unchanged. References to this **Translation Memory** may still exist in certain folders. Update the folder properties so that the deleted **Translation Memory** is no longer referenced.

Exporting a Translation Memory

To make a **Translation Memory** available to another system or user, you can export it to create an external **Translation Memory**. An external **Translation Memory** is in an SGML-based data exchange format that you can use to pass your **Translation Memory** data to users who do not work with **OpenTM2**.

Exporting a Translation Memory can also be part of the conversion of a Translation Memory into Unicode. To convert a Translation Memory into Unicode, you must first export it from OpenTM2 and then import it again. In both cases, you select **SGML UTF-16** as export format or import format.

You can also export the memory using the Translation Memory exchange format TMX.

Prerequisites

The **Translation Memory** must exist.

Calling sequence

Select:

1. The **Translation Memory** to be exported
2. **Export...** from the **File** menu

The "Export Translation Memory" window is displayed.

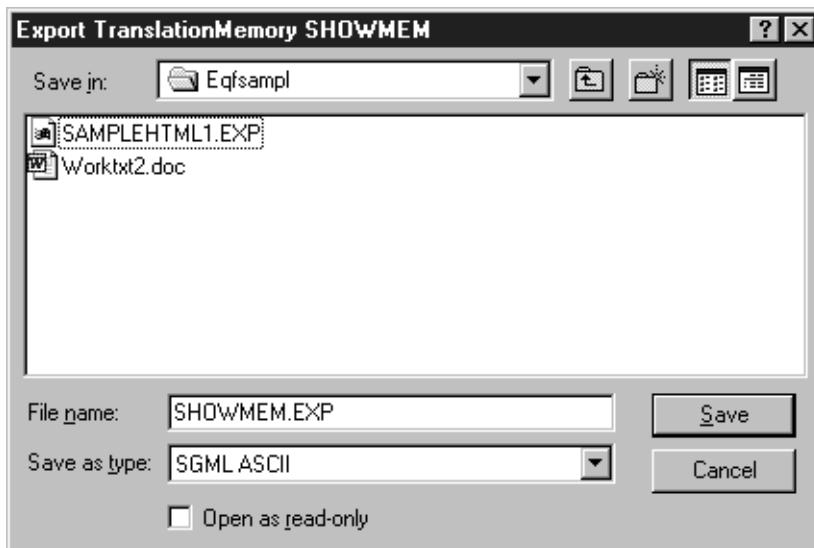


Figure 91. Export Translation Memory window

Options and parameters

Save in

Select the drive and directory to which the **Translation Memory** is to be exported.

File name

Contains the name of the **Translation Memory** to be exported. You can overwrite it with a new name.

Save as type

Select the format in which the Translation Memory is to be stored. Select **SGML UTF-16** if the export is part of your conversion of the Translation Memory into Unicode. Select **TMX (UTF-16)** or **TMX (UTF-8)** to export the memory in the TMX format.

Click **Save** to begin exporting the **Translation Memory**.

If you selected a diskette drive and the data volume of the **Translation Memory** exceeds the capacity of the diskette, **OpenTM2** informs you that it cannot export the **Translation Memory** in this way. You then must export the **Translation Memory** in a folder. For more information, see [Exporting a folder](#).

Results

The Translation Memory is exported as a file that you can pass to other users or systems. The default file extension is EXP for SGML format export and TMX for TMX format export. The internal Translation Memory remains unchanged.

Creating an Initial Translation Memory

reusingexisting translations Initial Translation Memory (ITM)creating creatingInitial Translation Memory (ITM) databases commandsEQFITM translationsreusing existing ones You can generate a Translation Memory already before you start translating a document in **OpenTM2**. In this way, you can immediately benefit from having a filled **Translation Memory** when you start translating, for example, updated versions of existing documents.

Such a **Translation Memory** is called an **Initial Translation Memory** (ITM). **OpenTM2** lets you create an Initial Translation Memory based on existing translations—both a source file, containing the original document, and a target file, containing the corresponding translated document, must be available.

OpenTM2 scans both files to find the corresponding target segment for each original segment, that is, it *aligns* the segments of the two files. You are recommended to check the combinations of segments afterwards.

When you use an Initial Translation Memory during a translation, the proposals in the "Translation Memory" window are prefixed with [m] (created by machine). **Translation Memory** matchesmachine-generated matchesmachine-generated Such *machine* proposals are treated as fuzzy matches, this means that they are not used during automatic substitution.

To generate an Initial Translation Memory, you can choose between two methods. You can start generating an Initial Translation Memory:

From the command area

In this case you must type the EQFITM command and all required parameters in your system's command area and press Enter. For a complete description of the syntax of this command, see "Creating an Initial Translation Memory from the command line".

From the Initial Translation Memory icon

In this case you must double-click the **Initial Translation Memory Tool** icon in the " OpenTM2" group window.

Note: In both cases the SGML memory is written from ITM in UNICODE format

The following sections describe how to:

- Generate an Initial Translation Memory
- Check the results of the alignment process (the process of combining source segments with their matching target segments)

Prerequisites

- The original documents and corresponding translations must be available as separate files.
- The **Translation Memory** that is subsequently to be used as an Initial Translation Memory must already be initialized (see "Creating a Translation Memory")
- The markup table for the document files must be the same for source and target documents.
- The language support for the source language and the target language must be installed.

Calling sequence

Type eqfitm in the command area of your system (next to the [C:\] prompt).

1. Press Enter to start the command.
2. Double-click the **Initial Translation Memory Tool** icon in the " OpenTM2" group window.

The "Create Initial Translation Memory" window is displayed:

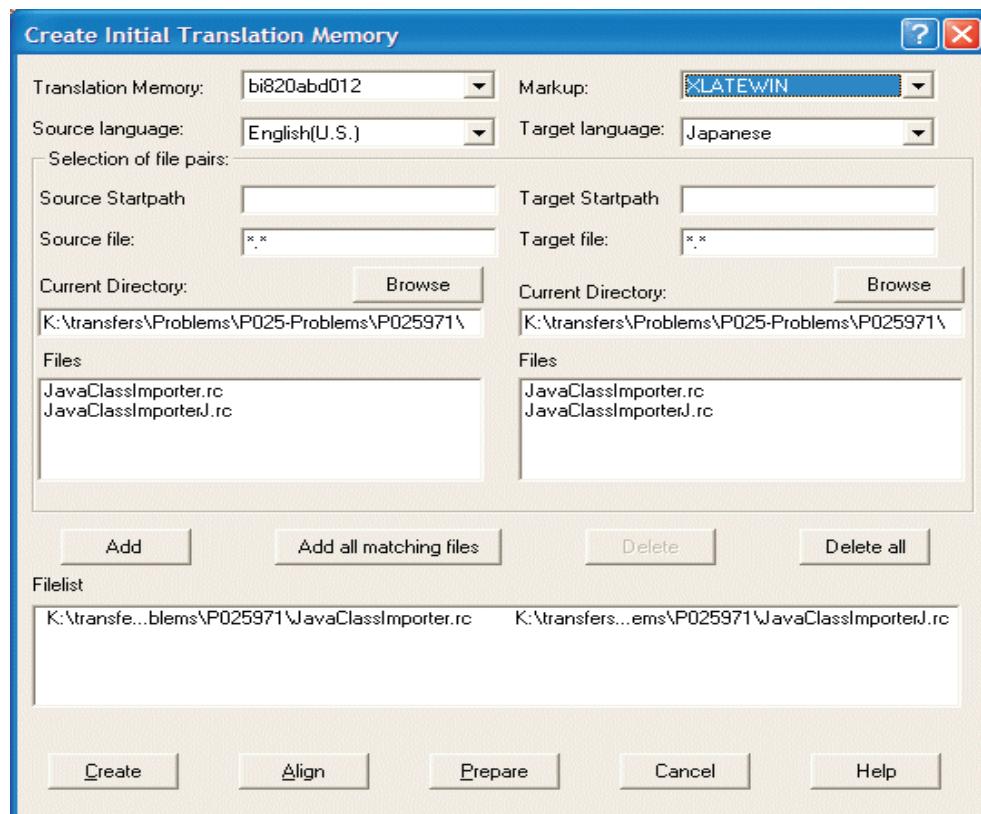


Figure 92. Create Initial Translation Memory window
[\[PIC\]](#)Figure shows Create Initial window

Options and parameters

Translation Memory

Select an existing **Translation Memory** to be filled with the matched segment pairs of existing translations (source segments and their matching target segments).

Markup

Select an existing markup table to be used for the file pairs you select.

Selection of file pairs

Define one file pair (source and target) at a time, then click **Add** to add the file pair to the **Filelist** list box. Then specify the next file pair or add all file pairs having the same name in the source file list and the target file list by clicking **Add all matching files**.

A source file is defined by the following parameters:

Source language

Select the language of the original document file from the list of available languages.

Source start path

Specify the path information that you do **not** want to become part of the document name when the original document is stored in the **Initial Translation Memory**. For example, if your source file is stored in e:\tm\project\english and you do not want e:\tm\project to become part of the name under which it is stored, specify e:\tm\project in this field.

The path you specify here can differ from the target start path. However, if you specify a source start path, you must also specify a target start path.

Source file

You can:

- Type the fully qualified file name in this field.
- Type only the file name but select the location of the file using the **Browse** button.
- If you first specify the file location using the **Browse** button or by typing in the location into the **Current Directory** field, select the file name from the **Files** list box.

Current directory

This field shows the drive and path currently selected, you can modify the path directly or you can click **Browse** to display the "Browse for Folder" window on which you can select the appropriate path.

Files

Contains a list of all the files in the current directory. Select the source file. The target file is defined similar to the source file:

Target language

Select the language of the target document from the list of available languages.

Target start path

Specify the path information that you do **not** want to become part of the document name when the target document is stored in the **Initial Translation Memory**. For example, if your target file is stored in `e:\tm\project\german` and you do not want `e:\tm\project` to become part of the name under which it is stored, specify `e:\tm\project` in this field.

The path you specify here can differ from the source start path. However, if you specify a target start path, you must also specify a source start path.

Target file

If the **Current Directory** field contains the correct drive and path name, either select the source document from the **Files** list box or type it directly into this field. Otherwise, first specify the correct drive and path information in the **Current Directory** field.

Current directory

This field shows the drive and path currently selected, you can modify the path directly or you can click **Browse** to display the "Browse for Folder" window on which you can select the appropriate path.

Files

Contains a list of all the files in the current directory. Select the target file.

Filelist

Displays file pairs selected for subsequent processing. All files must have the same markup. To change the selection of file pairs, use one of the following buttons:

Add

To save the specified file pair, click this button. The names of the source file and of the target file are added to the **Filelist** list box.

Add all matching files

To add all files having the same name in the source and target file list as file pairs.

Delete/Delete all

To remove a file pair, first select the file pair from the **Filelist** list box, then select **Delete**. To remove all file pairs, select **Delete all**.

Create

Starts the creation of the Initial Translation Memory and presents its contents on the screen for you to review. After you saved the Initial Translation Memory, the contained segments are treated as *machine* proposals, prefixed with [m]. To remove this prefix and enable the segments to be used during the automatic substitution process, use `EQFDMM.EXE`.

Align

When all file pairs have been added to the **Filelist** list box, click **Align** to begin filling the **Initial Translation Memory**. The aligning process starts and connects each source segment with a corresponding target segment. The matched segments are written to the selected Translation Memory (prefixed by [m]). Mind that these segment connections have not been checked yet. As it can take some time to join the document file pairs together, a window is displayed showing the progress of the operation.

Prepare

Same as **Create...** but no visual presentation of the Initial Translation Memory contents. The selected file pairs are prefixed by p. You can use this option to create the Initial Translation Memory but view it later.

Cancel

Stops the interaction with this window and returns you to the starting point. Any modifications you have made are not saved.

Help

Gives information to help you interact with the current window.

When the generation of the Initial Translation Memory is completed, a message window appears containing the number of paired segments. If you pressed the **Create...** button, the contents of the Initial Translation Memory is presented in a window on the screen. How you work with this window is described in [Revising an Initial Translation Memory](#).

Results

The Initial **Translation Memory** is filled with matching source and target segments, using the file pairs you selected.

It is recommended that you check the correctness of the matches made by **OpenTM2**. You can do this before you start translation work with this **Translation Memory** as described in [Revising an Initial Translation Memory](#), or online using the Initial Translation Memory specific editor (see [Revising an Initial Translation Memory](#)). This editor is invoked by selecting the **Create** button in the "Create Initial Translation Memory" window.

If you start to use the Initial **Translation Memory** without any further checking, it is recommended to treat the *machine-generated* matches (prefixed by [m]) in the same way as fuzzy matches (prefixed by [f]). When you copy these proposals into your translation, check whether they need to be adapted and change them where necessary.

If you choose to use automatic substitution during analysis, *machine-generated* matches are not used.

Importing a Translation Memory

importing Translation Memory databases **Translation Memory importing** If you receive an external **Translation Memory**, you can import it into **OpenTM2**.

During the import, you can also merge the data from an external **Translation Memory** into a **Translation Memory** that already exists in **OpenTM2**. **Translation Memory Unicode enablement**

Importing a Translation Memory can also be part of the conversion of a Translation Memory into Unicode. To convert a Translation Memory into Unicode, you must first export it from OpenTM2 and then import it again. In both cases, you select **SGML Unicode** as export or import format.

Prerequisites

- The **Translation Memory** to be imported must exist.
- If you want to merge the data from the external **Translation Memory** into an existing **Translation Memory**, both **Translation Memory databases** must have the same source language.

Calling sequence

Select:

1. The "Translation Memory List" window from the main window
2. Optionally the **Translation Memory** into which to import the external **Translation Memory**
3. **Import...** from the **File** menu

The "Import Translation Memory" window (see [Figure 93](#)) is displayed.

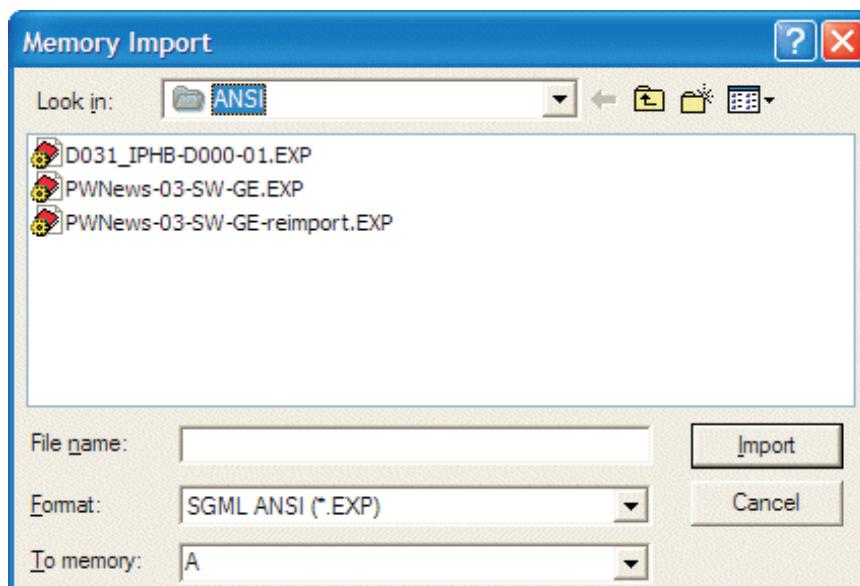


Figure 93. Import Translation Memory window
[PIC]Figure shows Import window

Options and parameters

Look in

Select the directory where Translation Memory to be imported is currently stored.

File name

Enter the file name of the Translation Memory to be imported or select one or more entries in the list box.

Format

Select the format of the Translation Memory to be imported

- SGML ANSI for the import of external memory databases in the SGML (.EXP) format in ANSI encoding
- SGML ASCII for the import of external memory databases in the SGML (.EXP) format in ASCII encoding
- SGML UTF16 for the import of external memory databases in the SGML (.EXP) format in Unicode (UTF-16) encoding
- TMX for the import of external memory databases in the TMX (Translation Memory Exchange) format
- TMX (Trados) for the import of external memory databases in the TMX (Translation Memory Exchange) format which have been exported using the Trados translation tool, for this special import format any RTF tagging contained in the imported segment data is removed

To memory

Contains the name of the **Translation Memory** to be imported. You can overwrite this name or select one from the list box. If you specify a **Translation Memory** that does not exist yet, the "New Translation Memory" window is displayed for you to create it (see [Creating a Translation Memory](#)).

If you select or specify an existing **Translation Memory**, the contents of the **Translation Memory** to be imported are merged into it.

When you have specified your input, click **Import**.

For information on importing **Translation Memory databases** from other OpenTM2 products, see [Exchanging data with other OpenTM2 products](#).

Results

The **Translation Memory** data is imported into the selected target **Translation Memory** and you can begin to work with it. The external **Translation Memory** remains unchanged.

Merging Translation Memory databases

Translation Memory merging If several translators are working on a large document, each of them uses and updates a separate **Translation Memory**.

For the translation of subsequent releases of the document, it is useful to merge these to create a single consolidated **Translation Memory**.

You can merge **Translation Memory databases** in two ways:

- While importing an external **Translation Memory** (see ["Importing a Translation Memory"](#)).
- By merging two internal **Translation Memory databases**

The merging of two **Translation Memory databases** is described using two sample names, TMEM01 and TMEM02.

Prerequisites

- The two **Translation Memory databases** to be merged must exist.
- They must have the same source language.

Calling sequence

Select:

1. The "Translation Memory List" window
2. The **Translation Memory** to be merged, for example, TMEM01
3. Select **Merge...** from the **File** menu

The "Merge Translation Memory" window (see [Figure 94](#)) is displayed.

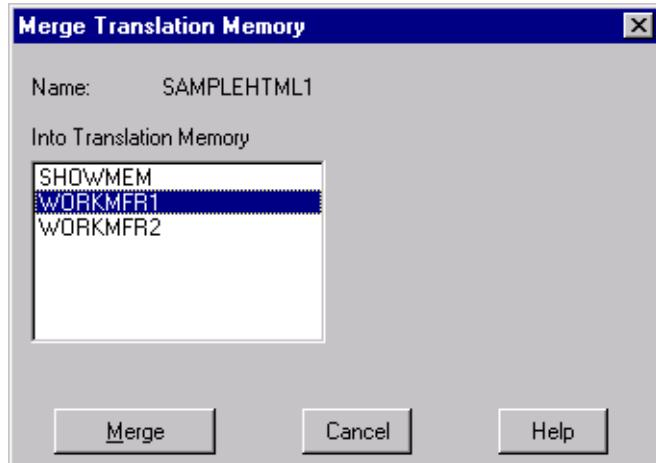


Figure 94. Merge Translation Memory window

Options and parameters

The **Into Translation Memory** list box displays all available **Translation Memory databases**. Select the **Translation Memory**, for example TMEM02, into which to merge the previously selected **Translation Memory** TMEM01.

When you have selected the target **Translation Memory**, click **Merge** to begin merging the **Translation Memory databases**.

Results

The contents of TMEM01 is merged into the target **Translation Memory** TMEM02. The **Translation Memory** TMEM01 remains unchanged. You can begin to use the extended **Translation Memory** TMEM02.

Archiving a Translation Memory

Translation Memory archiving When you have finished a translation project you can archive the translated documents in another or a new **Translation Memory**. You then have a clean **Translation Memory** without redundancies.

Prerequisites

At least one segment of the document or documents you want to archive must have been translated.

Calling sequence

Select:

1. The "Folder List" or "Document List" window, depending on whether you want to archive a single or all documents in a folder.
2. The folder or document.
3. **Build Archive Translation Memory...** from the **Utilities** menu.

The "Build Archive Translation Memory" window (see [Figure 95](#)) is displayed.

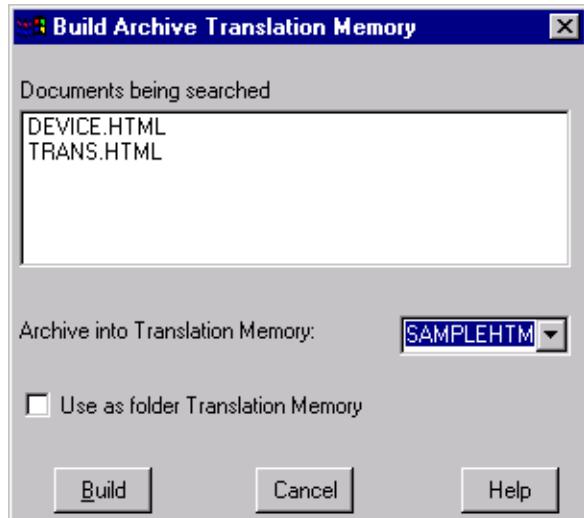


Figure 95. Build Archive Translation Memory window

Options and parameters

Documents being searched

This box lists the documents that are searched for translated segments to be included in the **Translation Memory**.

Archive into Translation Memory

Select an existing **Translation Memory** or type the name for a new **Translation Memory** in which the documents are to be archived.

Use as folder Translation Memory

Select this option if you want to use the **Translation Memory** as the new folder **Translation Memory**.

Results

The documents are stored in the specified **Translation Memory**. If you specified an existing **Translation Memory**, you are asked whether you want to overwrite its contents or merge the new entries with the existing ones.

Organizing a Translation Memory

Translation Memory organizing organizing Translation Memory databases Translation Memory removing obsolete segments Translation Memory repairing defective ones Translation Memory upgrading backlevel versions Occasionally OpenTM2 prompts you to organize a Translation Memory. A Translation Memory must be organized to delete obsolete segments, to repair a defective Translation Memory, or to upgrade a backlevel Translation Memory.

Prerequisites

The **Translation Memory** to be organized must exist.

Calling sequence

Select:

1. The "Translation Memory List" window
2. The **Translation Memory** to be organized
3. **Organize** from the **File** menu

Organizing begins immediately. As it may take some time to organize a large **Translation Memory**, a window is displayed showing you the progress.

Options and parameters

None.

Results

The **Translation Memory** is now organized and the new number of segments in the **Translation Memory** is displayed. You can continue to use it.

Revising an Initial Translation Memory

revising Initial Translation Memory postediting Initial Translation Memory Initial Translation Memory (ITM) revising Initial Translation Memory (ITM) postediting editing Initial Translation Memory To increase the reliability of the automatically generated Initial Translation Memory, it is necessary to check the correctness of the relation between source and target segments.

With the Initial Translation Memory editor you can see the contents of the Initial Translation Memory at a glance to verify the correspondence between source and target segments. When a target segment does not correspond to a source segment, you can perform the necessary changes.

The original segments are displayed in the "Source" window, the corresponding translations in the "Target" window. The segment pairs are synchronized with each other. That means that a target segment is displayed at the same level on the screen as the source segment and with the same background color. The following describes how to work with the Initial Translation Memory editor.

Prerequisites

The "Create Initial Translation Memory" window must be invoked and filled in.

Calling sequence

Select:

- The **Create** button from the "Create Initial Translation Memory" window.

The "Edit Initial Translation Memory" window is displayed. It consists of two windows. One window shows source segments, the other window shows the corresponding target segments.

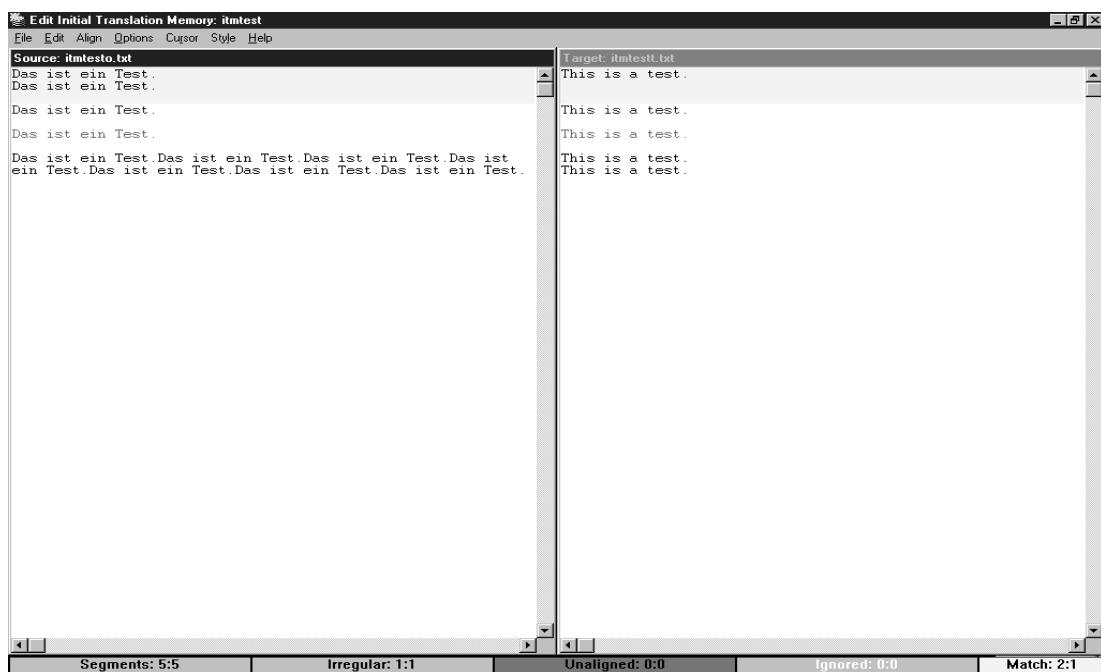


Figure 96. Edit Initial Translation Memory window
[PIC]Figure shows Edit Initial Translation Memory window

Options and parameters

For information on how to work with the Initial Translation Memory editor see [The Initial Translation Memory editor](#).

Results

You are now ready to verify the correctness of the automatically generated pairs of source segments and their translated equivalents. When you leave the editor the segment pairs are stored in the Translation Memory.

The Initial Translation Memory editor

On the menu bar of the "Edit Initial Translation Memory" window you can select the following items:

File

To save changes in the "Initial Translation Memory" window and to exit.

Edit

To find and replace, split and join, cut and paste, delete, and do other operations on the text.

Align

To manipulate the relation between source and target segments.

Options

To change the view of the "Source" and "Target" windows and to modify colors and fonts.

Cursor

To move and change the cursor position in the "Source" and "Target" windows, to mark segments, and to set bookmarks.

Style

To change the display of markup information.

Help

To get information that helps you interact with the current window.

File menu

Menu or key	Function
Save (F2)	Saves the current segment connections. If you have selected another pair of files, the segment connections resulting from these files will be displayed next.
Quit (F3)	Quits the "Edit Initial Translation Memory"

Menu or key	Function
	window without saving changes.
End-Save (F4)	Quits the "Edit Initial Translation Memory" window and saves any changes to the selected Translation Memory.
Toggle window (Ctrl+t)	Activates the opposite window ("Source" or "Target").

Edit menu

Menu or key	Function
Find and replace... (Ctrl+f)	Locates a term in the source or target document, depending on where the cursor is located, and changes it as specified. You can only change a term if the Initial Translation Memory can be edited.
Cut (Ctrl+x)	Copies a marked text section to the system clipboard and removes text afterwards.
Copy (Ctrl+c)	Copies a marked text section to the system clipboard for later pasting.
Paste (Ctrl+v)	Inserts text from the clipboard into the active segment at the current cursor position.
Undo	Restores the original state as it was before an action was performed.
Split line	Splits the line at the current cursor position.
Join line	Combines the current line with the following line.
Toggle initial caps	Changes the initial caps to small initials.
Add an abbreviation (Ctrl+a)	Adds the term where the cursor is positioned to the list of abbreviations associated to the current source language.

Align menu

Menu or key	Function
Connect segments (Ctrl+o)	Relates the marked source segment to the marked target segment.
Delete connection (Ctrl+d)	Removes a connection between a source segment and a target segment. Either of the segments must be marked.
Ignore segment (Ctrl+i)	Removes a marked segment from the ITM which you do not want to be connected with another segment, or that does not have a match in the opposite file. The ignored segment remains visible but gets a different background color.
Undo Ignore (Ctrl+u)	Makes a previously ignored segment available for connection with a matching segment in the opposite file. Note that the segment must be marked before you use .
Join segments (Ctrl+j)	Combines the currently active segment with the following one.
Split joined segments (Ctrl+s)	Splits a previously joined segment at the current cursor position.

Options menu

Menu	Function
Colors...	Lets you define the colors for the segments in the "Source" and "Target" windows.
Fonts...	Lets you set the font and font size for the "Source" and "Target" windows in the "Set Fonts" window.
Arrange	Changes the arrangement of the "Source" and "Target" windows. You can arrange them (the target document is positioned next to the source document on the right half of the screen) or (the target document is positioned below

Menu	Function
	the source document on the lower half of the screen).
Parallel	Positions the segments in the "Source" window parallel to their matches in the "Target" window. That means that lines may be inserted in either the "Source" or the "Target" window to achieve that a segment is presented at the same position as its counterpart.
Automatic linewrap	Wraps the text at the end of a line.
Visible white spaces	Replaces blanks in the text with a dot and displays a ¶ sign at the end of a segment or in an empty line to make them visible.

Cursor menu

Menu	Function
Next connection (Ctrl+Shift+Down)	Moves the cursor to the next connected segment pair.
Previous connection (Ctrl+Shift+Up)	Moves the cursor to the previous connected segment pair.
Next unaligned (Alt+Down)	Moves the cursor to the next segment for which no matching segment has been found. This can be either in the same or in the opposite window.
Previous unaligned (Alt+Up)	Moves the cursor to the previous segment for which no matching segment has been found. This can be either in the same or in the opposite window.
Next irregular (Shift+Alt+Down)	Moves the cursor to the next and marks it. An irregular match can be one of the following: <ul style="list-style-type: none"> • A 1:2 match, where one source segment is connected to two target segments • A 2:1 match, where two source segments are connected to one target segment • A 2:2 match, where two source segments are connected to two target segments • An unaligned sentence (the default color is red) • A sentence that is ignored (the default color is gray)
Previous irregular (Shift+Alt+Up)	Moves the cursor to the previous irregular match and marks it.
Synchronize (Ctrl+Enter)	Positions the current segment and its match in the opposite window beside one another at the same level.
Mark segment (Ctrl+m)	Marks the segment where the cursor is positioned. A segment must be marked to connect and to remove it.
Unmark segment (Ctrl+n)	Removes the marking from a segment. Only one segment in a window can be marked.
Go to line...	Enables you to specify the number of the line to which you want to move your cursor.
Query line	Informs you in which line your cursor is located.
Set bookmark	Sets a bookmark at the current cursor position. This makes it easier for you to return to this point in the Initial Translation Memory at a later point of time. You can set several bookmarks, for example, at all those alignments that you want to check again later.
Go to bookmark	Moves the cursor to the position where you set a bookmark.
Clear bookmark	Removes a bookmark from the segment where the cursor is located.

Style menu

Menu	Function
Protect	The markup tags are shown but cannot be overwritten (see the example given in Figure 2).
Unprotect	The markup tags are shown and you can overwrite and change them.
Hide	The markup tags are not shown.
Shrink	All markup tags outside segments are not shown. A replacement character is shown instead (see the example given in Figure 2).
Compact	All markup tags are replaced with a replacement character to indicate the position of each tag (see the example given in Figure 2).
Compact+1	Markup tags are shown up to a length of 10 characters and cannot be changed. Longer markup tags are followed by three dots (...), for example [Style=@Out...]

The **status bar** at the bottom of the window displays the following information from left to right:

1. The total number of segments in the source window compared to the total number of segments in the target window.
2. The number of segments in the source window that have an irregular match in the target window compared to the number of segments in the target window that have an irregular match in the source window.
3. The number of segments in the source window that have no match in the target window compared to the number of segments in the target window that have no match in the source window.
4. The number of ignored segments in the source window compared to the number of ignored segments in the target window.
5. The relation of source segments to target segments. For example, **2:1** means that two source segments have been connected to one target segment.

Revising a Translation Memory

revising Translation Memory databases postediting Translation Memory databases Translation Memory revising Translation Memory postediting editing Translation Memory databases To improve the quality of your translations, you may wish to check the contents of a Translation Memory from time to time, and modify or remove translations that are stored there. Doing these changes directly in the Translation Memory lets you benefit from polished translations the next time you use this repository.

Therefore, OpenTM2 offers a Translation Memory that can be manipulated. You can browse its contents, make changes to existing translations, or delete complete entries from it. You can work on a Translation Memory contents in an editor window as if you were editing normal text. The original segments are displayed in the "Original" window, the corresponding translations in the "Translation" window.

You make your changes in the "Translation" window. Translation Memory files can be huge. To make it easier for you to find a certain translation, you have the choice to narrow down what you are looking for. For example, you can search for a certain translation segment, or for translations made within a certain time frame. The following describes how to open a Translation Memory in order to work with it.

Note:

Use this function with care. A Translation Memory that has grown over a period of time is an essential asset for all your translations. Therefore it is good practice to avoid any uncontrolled changes. There is no Quit or Undo function. All changes become effective immediately.

Prerequisites

The Translation Memory you want to work with must exist.

Calling sequence

Select:

1. The "Translation Memory List" window from the main window
2. The Translation Memory you wish to edit
3. **Open** from the **File** menu or double-click the Translation Memory name

You are first prompted with the "Translation Memory Editor" window to specify what you are looking for in the selected Translation Memory. For more information on how to work with this window see [Searching a translation](#). After you specified the search criteria, the "Edit Translation Memory" window (see [Figure 1](#)) is displayed. It consists of two windows. The "Original" window lets you view the original text, in the "Translation" window you can make your changes. By pressing Ctrl+Enter you can move from segment to segment.

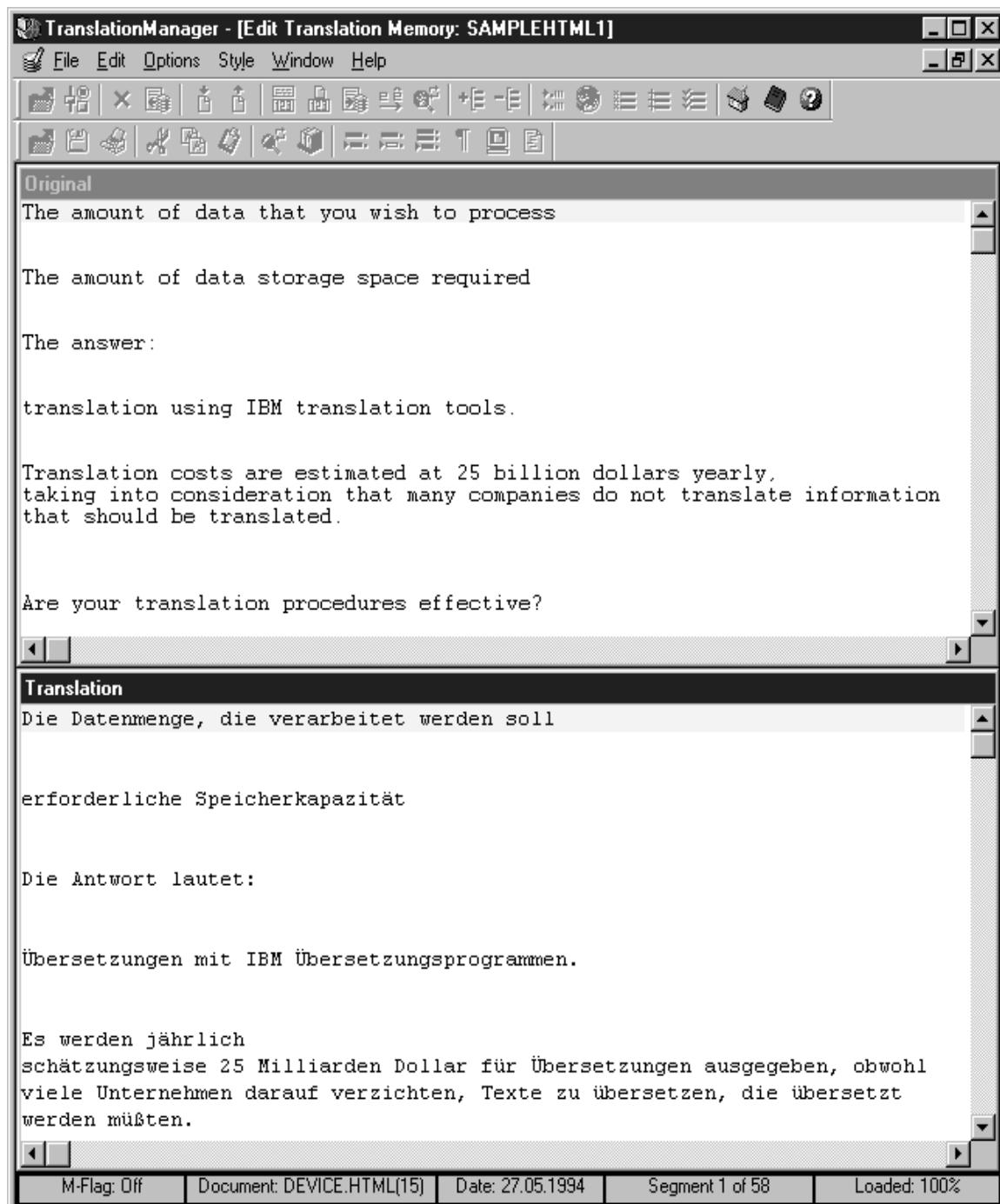


Figure 97. Edit Translation Memory window

Options and parameters

For information on how to use the Translation Memory editor functions, see [The Translation Memory editor](#).

Results

You open a Translation Memory to work on the translations that are stored in it. In the "Edit Translation Memory" window you can browse the original texts and revise the corresponding translations. If you need to change or remove translated text, you can do this in the "Translation" window where you are assisted by an editor. How you use this editor is explained in [The Translation Memory editor](#).

Searching a translation

Translation Memorysearching a translation Translation Memory files can be huge. You can search for a certain translation segment or for specific translations. In the "Translation Memory Editor" window you specify the individual search criteria. The **Look up...** button starts the search in the Translation Memory and displays the found Translation Memory contents in the "Edit Translation Memory" window. The following describes how to specify the search criteria.

Prerequisites

The Translation Memory you want to work with must exist.

Calling sequence

Select:

1. The "Translation Memory List" window from the main window
2. The Translation Memory you wish to edit
3. **Open** from the **File** menu

The "Translation Memory Editor" window (see [Figure 98](#)) is displayed.

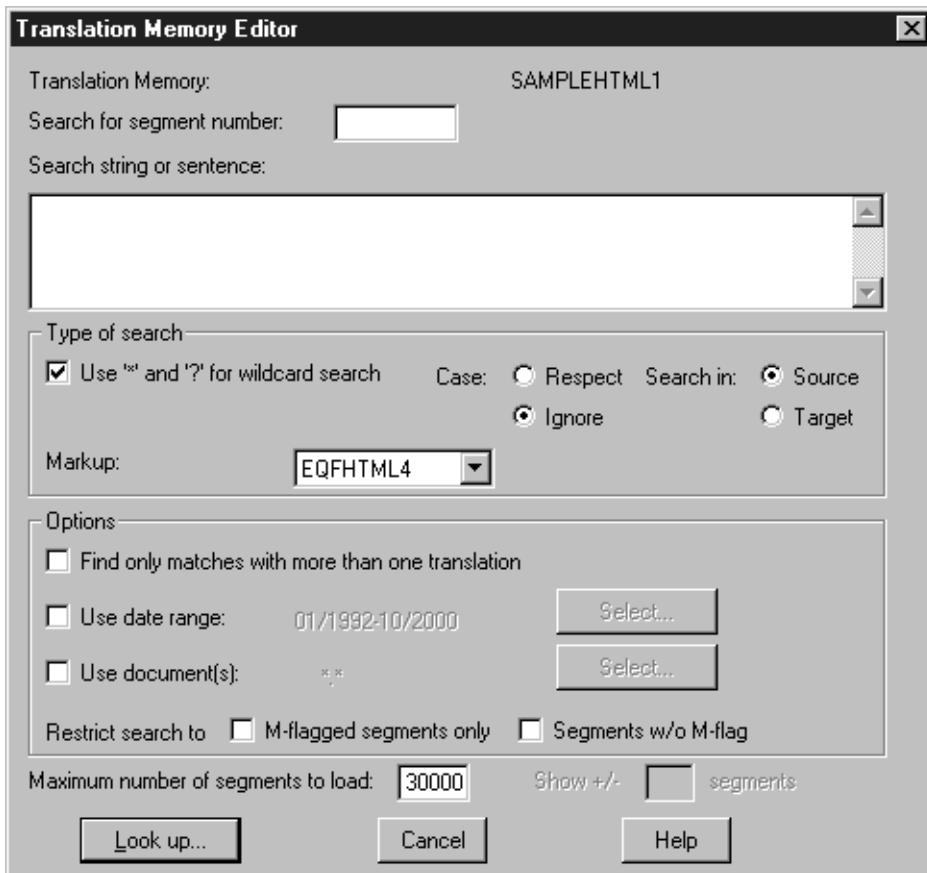


Figure 98. Translation Memory Editor window

Options and parameters

Search for segment number

Each segment in the **Translation Memory** has a number. It is displayed in the status bar of the "Edit Translation Memory" window as part of the document name. Instead of specifying the segment to be searched, or part of it, you can type its number here.

Search string or sentence

Type the term or sentence you are looking for. You can also use wildcard characters (**Use '*' and '?' for wildcard search**). For example, if you are looking for a product name beginning with "Star" and you are not sure about the product's exact name, you can type `Star*`.

Case

Specify whether the search for the requested text string is to be case-sensitive (**respect** case sensitivity). The default setting is **ignore** case sensitivity.

Search in

Specify whether the search is to be performed on the original entries (**Source**) or on the translations (**Target**). The default setting is **Source**.

Markup

Specify the markup language of the text you are looking for.

Find only matches with more than one translation

Specify whether you want to retrieve only original segments with more than one translation. If there is only one translation for an original segment, this segment will be ignored during the search in the Translation Memory and not displayed. Use this option to revise translation variants.

Use date range

Specify the period of time when translations were stored in the Translation Memory selected. Only the original segments of those translations that were added to the Translation Memory during the specified period are displayed in the "Edit Translation Memory" window. The default setting for the time span is January 1992 until the present time. Select **Select...** to alter this time span.

Use document(s)

Type the name of the document from which the translation is to originate. Select **Select...** to type the name of a document. You may use * as wildcard character.

Restrict search to

Specify whether you want to restrict your search to segments that were translated by *machine (m-flagged segments only)* or segments that were not translated by machine (**segments w/o m flag**).

Maximum number of segments to load

The segments are loaded into the editor in stages. Specify the maximum number of segments that you want to be loaded into the Translation Memory editor at one time. You can specify a number between 5 and 99999. The default setting is 99999.

Click **Look up...** to start the search.

Results

The Translation Memory is searched according to the search criteria specified in the "Translation Memory Editor" window. The found matches are loaded into the editor and displayed in the "Edit Translation Memory" window where you can revise them. How you revise Translation Memory entries is described in [The Translation Memory editor](#).

The Translation Memory editor

To revise the entries in a Translation Memory, you are assisted by an editor. The functions of this editor support you when changing the Translation Memory entries.

The editor consists of two windows. The "Original" window shows original segments, the "Translation" window shows the corresponding translations. By pressing Ctrl+Enter you can move from segment to segment.

From the menu bar of the "Edit Translation Memory" window you can select the following items:

File

To save changes, to exit the editor, to load more segments, to switch between "Translation" and "Editor" window.

Edit

To find and replace, to split and join, to cut and paste, to delete, and do other operations on the text.

Options

To change colors and fonts, and to arrange the editor windows.

Style

To change your view of markup tags.

File menu

Menu or key	Function
Next	Loads another collection of segments into the editor if more segments are found in the Translation Memory than the number set in the "Translation Memory Editor" window.
Previous	Loads the previous collection of segment pairs into the editor.
New query (F5)	Saves any changes made in the Translation Memory. Returns you to the "Translation Memory Editor" window.
Quit (F4)	Saves any changes made in the Translation Memory editor and returns you to the OpenTM2 main window.
Toggle windows (Ctrl+t)	Activates the opposite window ("Original" or

Menu or key	Function
	"Translation").

Edit menu

Many of the editor functions are tied to the "Translation" window only. If a function on the **Edit** menu cannot be performed in the "Original" window, the function is disabled whenever the "Original" window becomes active. If a keystroke editor function does not apply to the active window, the system beeps.

Menu or key	Function
Find and replace... (Alt+F6)	Locates a term and replaces it as specified. Works in "Translation" and "Original" window. Changes in the original text are not allowed.
Cut	Copies a marked text section to the system clipboard and removes text afterwards. Works in "Translation" window only.
Copy	Copies a marked text section to the system clipboard for later pasting. Works in "Translation" and "Original" window. Text cannot be pasted into "Original" window though.
Paste	Inserts text from clipboard into active segment at current cursor position. Works in "Translation" window only.
Undo	Restores the original state as it was before an action was performed. Works in "Translation" window only.
Unmark block (Alt+m)	Removes marking from a text section. Works in "Translation" and "Original" window.
Split line (Alt+s)	Splits line at current cursor position. Works in "Translation" window only.
Join line (Alt+j)	Combines current line with following line. Works in "Translation" window only.
Delete	Lets you choose either of the following options: <ul style="list-style-type: none"> • Current segment (Ctrl+d) Removes only the currently active segment from the Translation Memory. • All segments Removes all segments from the Translation Memory (those that were loaded in the editor). Works in "Translation" and "Original" window.
Set m flag	Lets you choose one of the following options: <ul style="list-style-type: none"> • Delete one m flag Deletes the m flag contained in the current segment. • Delete all m flags Deletes all m flags contained in the Translation Memory. • Set one m flag Displays the m flag for the current segment. • Set all m flags Displays the m flags for all machine translations in the Translation Memory that are currently loaded into the editor.

The following key combinations let you manipulate text in both the "Original" and "Translation" windows. Note that the two windows are synchronized with each other. That means, whenever you activate a segment in the "Translation" window, it becomes active in the "Original" window too, and vice versa. The matching segment in the opposite window is presented beside the active one at the same level.

Keys	Function
Ctrl+Shift+Left	Marks previous word from cursor position to beginning of word.
Ctrl+Shift+Right	Marks next word from cursor position to end of

Keys	Function
	word including the blank space.
Ctrl+Insert	Copies marked block to system clipboard.
Ctrl+Delete	Deletes all characters from current cursor position to end of line in active segment. Works in "Translation" window only.
Ctrl+Home	Moves cursor to first segment in window.
Ctrl+End	Moves cursor to last segment in window.
Ctrl+left	Moves cursor to beginning of previous word.
Ctrl+right	Moves cursor to beginning of next word.
Ctrl+Enter	Activates next segment or segment where cursor is positioned.
Ctrl+t	Toggles between "Original" window and "Translation" window.
Shift+left	Extends text marking by one character to the left.
Shift+right	Extends text marking by one character to the right.
Shift+Up	Extends text marking to the previous line.
Shift+Down	Extends text marking to the next line.
Shift+Insert	Inserts text from clipboard at current cursor position in active segment.
Alt+F6	Locates a term and changes it as specified. Works in "Translation" window and "Original" window. Changes in the original text are not allowed.
Alt+Delete	Deletes all characters from current cursor position to end of active segment. Works in "Translation" window only.
Alt+j	Joins current line with following line. Works in "Translation" window only.
Alt+s	Splits line at current cursor position. Works in "Translation" window only.
Alt+n	Loads next collection of segments.
Alt+p	Loads previous collection.
Alt+m	Removes marking from a text section.
Up	Moves cursor one line up.
Down	Moves cursor one line down.
Right	Moves cursor one character right.
Left	Moves cursor one character left.
Home	Moves cursor to beginning of current line.
End	Moves cursor to end of current line.
Page Up	Moves text one page up.
Page Down	Moves text one page down.
Delete	If a text section has been marked, the marked text is removed. If no text has been marked, the character where the cursor is positioned is removed.
Backspace	Moves cursor back by one character in the active segment and deletes this character.
Insert	Switches from insert mode to overtype mode and vice versa. When you insert text in insert mode, the existing text is moved to the right to make room for the new text. If you want to type over the existing text, switch to overtype mode.
F3	Quits the Translation Memory editor and returns you to the workbench.
F4	Saves any changes made in the Translation Memory and returns you to the "Translation Memory Editor" window.
Enter	Moves to next line.
Double-click left mouse button	Marks entire segment beginning from the cursor position.
Double-click right mouse button	Removes marking from a segment.

Options menu

Menu	Function
Colors...	Lets you change foreground and background colors. Note that if you change colors in the "Translation" window or the "Original" window, it will change colors in the translation editor windows accordingly. See also to learn how to work with the "Set Colors" window.
Fonts...	Lets you change fonts and their sizes. Note that if you change a font or a font size in the "Translation" window or the "Original" window, it will change the font or font size in the translation editor windows accordingly. See also to learn how to work with the "Set Fonts" window.
Arrange	Lets you arrange windows. Horizontal positions the "Translation" window below the "Original" window. positions the "Translation" window next to the "Original" window on the right half of the screen.

Style menu

Menu	Function
Protect	Markup tags cannot be overwritten.
Unprotect	Markup tags within a currently active translation can be overwritten.
Hide	Markup tags are not shown (they are there, but invisible).
Compact	Markup tags are replaced by a special character indicating the position of the tag.

The **status bar** at the bottom of the editor window displays the following information (from left to right):

M flag

Shows whether or not the **Translation Memory** contains any m flags (**On** or **Off**).

Document

Displays the name of the document where the active translation segment originates from and the line number of the segment.

Date

Displays the date at which the translation was added to the Translation Memory.

Segment n of m

Displays the number of the currently active segment (*n*) and the total number of segments loaded into the editor (*m*).

Loaded

Displays how much of the Translation Memory has been loaded into the editor.

Revising external Translation Memory databases

An external **Translation Memory** contains segments in their original sequence. That is, the segments are placed in an external **Translation Memory** in the sequence in which they occur in the documents.

You can edit the external **Translation Memory** with an editor of your choice as follows:

- Remove all segments that have no corresponding source or target segment.
- To show that a segment pair has been checked for correctness by a human translator and that it is no longer a machine-generated match, change the character that indicates machine translation in the external **Translation Memory**. To see where this character is located in the segment control information of the external **Translation Memory** and how to change it, refer to "Format of an extarnel Translation Memory".

Viewing the properties of a Translation Memory

Translation Memory displaying properties Translation Memory properties The properties of a **Translation Memory** are:

Memory Name

The name of the **Translation Memory**.

Full Memory Name

The name of the **Translation Memory** including its location.

Description

The description of the **Translation Memory**.

Source language

The source of the **Translation Memory**, this means, the language of the terms (headwords).

Drive

The drive where the **Translation Memory** resides.

Last update

The date and time of the last update of the **Translation Memory**.

Size of index file

The total size of the **Translation Memory**, in bytes.

Different Segments

The total number of different segments stored in the **Translation Memory**, which have different translations, which are from different languages or which have different markup tables.

Markup

A list of all markup tables used by the segments of this **Translation Memory**.

Documents

A list of the documents for which segments are stored in this **Translation Memory**.

Prerequisites

The **Translation Memory** must exist.

Calling sequence

Select:

1. A **Translation Memory** from the "Translation Memory List" window.
2. **Properties Summary...** from the **File** menu

Results

A document in HTML format is displayed containing a summary of the **Translation Memory** properties (see [Figure 99](#)) is displayed.

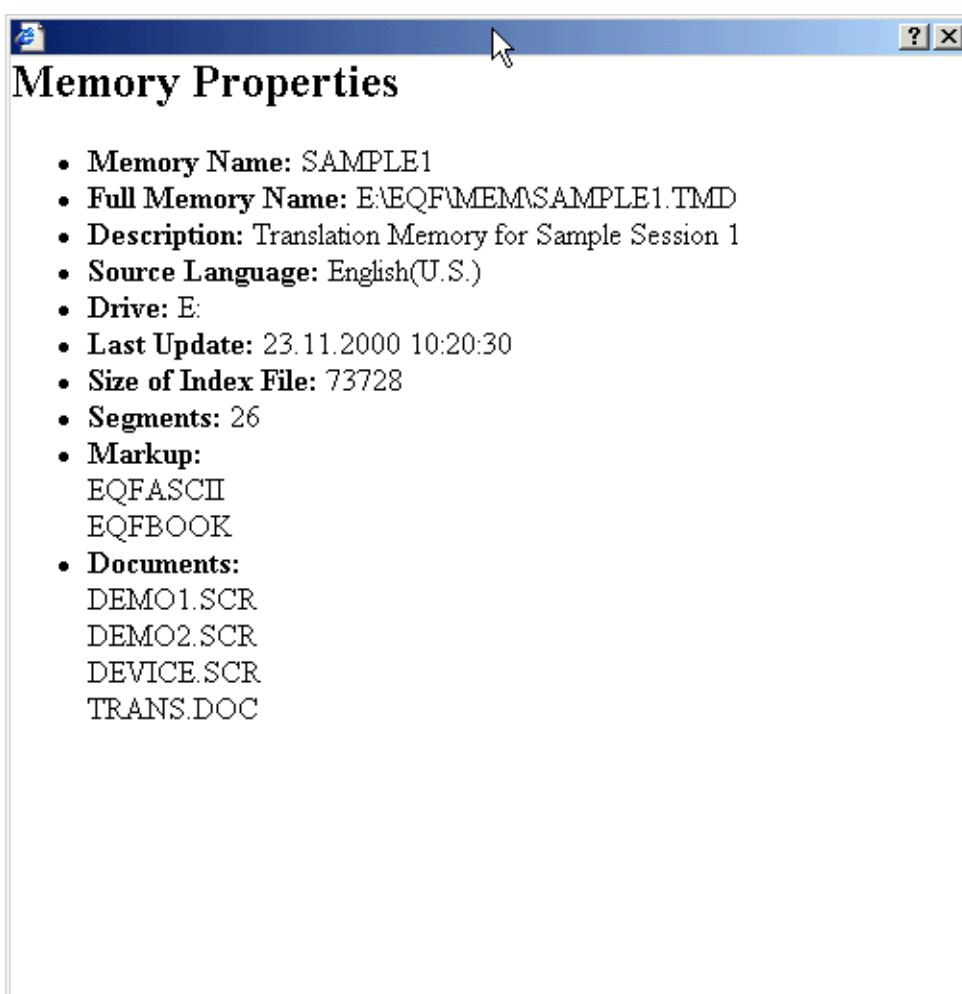


Figure 99. Memory Properties document

If you want to view more properties or change the properties of a Translation Memory, refer to [Changing the properties of a Translation Memory](#).

Changing the properties of a Translation Memory

Translation Memory changing properties All properties of a **Translation Memory** can be viewed, but only the description can be changed.

Prerequisites

The **Translation Memory** must exist.

Calling sequence

Select:

1. The "Translation Memory List" window
2. The **Translation Memory** from the list of **Translation Memory databases**
3. **Properties...** from the **File** menu

This takes you to the "Translation Memory Properties" window (see [Figure 100](#)).



Figure 100. Translation Memory Properties window

Options and parameters

The "Translation Memory Properties" window lists the characteristics of the selected **Translation Memory**. Only the **Description** field can be changed.

Name

The name of the **Translation Memory**.

Short name

The short name of the Translation Memory.

Description

The description of the **Translation Memory**. It can be up to 40 characters long.

Location

Can be **local** (only one translator can use it at a time) or **shared** (several translators can use it simultaneously).

Disk drive

The drive where the **Translation Memory** resides.

Source language

The source of the **Translation Memory**, this means the language of the terms (headwords).

Different Segments

The total number of different segments stored in the **Translation Memory** which have different translations, which are from different languages or which have different markup tables.

File size

The total size of the **Translation Memory**, in bytes.

Last update

The date and time of the last update of the **Translation Memory**.

Documents

A list of the documents for which segments are stored in this **Translation Memory**.

Markups

A list of all markup tables used by the segments of this **Translation Memory**.

Results

If you select **Change**, the **Translation Memory** description is changed according to your specification. Otherwise, it remains unchanged. You are returned to the previous window.

Viewing the details of a Translation Memory

In the "Translation Memory List" window, you can display additional information about a **Translation Memory**. You can define which details you want to be displayed and in which order.

The **Translation Memory** details are:

Name

The name of the Translation Memory.

Description

The description of the Translation Memory.

Size

The size of the Translation Memory (number of bytes).

Drive

The drive on which the Translation Memory resides.

Owner

Shows who created the Translation Memory if it is a shared one. If it is a local Translation Memory, n/a is displayed.

Source Language

The language of the source segments.

You can toggle between two display modes:

- Display the Translation Memory names only
- Display selected details

You can change the arrangement of the **Translation Memory** details in the "Change Details" window described in [Viewing and changing the details of a list item](#).

Working with dictionaries

A dictionary is a database that contains terms, their translations, and other related information.

OpenTM2 provides a set of bilingual dictionaries with general vocabulary for English, German, French, Spanish, and Italian.

You need dictionaries to look up the translations of individual terms or have these terms displayed automatically in the "Dictionary" window during translation.

A *dictionary entry* contains all data relating to a term (or *headword*). The dictionary entry structure describes which data elements can be stored for a term in a dictionary.

OpenTM2 provides a general dictionary structure that you can use as a model for building a dictionary serving your own purposes. Headword and Translation are mandatory fields. Certain entry fields, for example Synonym and Related term, are reserved for search criteria available during lookup in a dictionary.

OpenTM2 can handle dictionaries with simple and complex structures.

The general dictionary entry structure in **OpenTM2** is divided into four embedded levels to contain, for example, verb and noun entries, different meanings (senses), and translation variants.

In **OpenTM2** a dictionary can have two formats:

- An internal dictionary is a dictionary that exists in **OpenTM2**.
- A dictionary in external format is an SGML-based data file. SGML is the data exchange format required for importing dictionary data into **OpenTM2**. When a dictionary in the internal format is exported, it is automatically converted into SGML. When you import an external dictionary, it is automatically converted to the **OpenTM2** format.

All dictionary data in other text-processor or database formats needs to be converted into SGML by, for example, a program or database macro.

You can create your own dictionaries from existing terminology imported into **OpenTM2**, or from scratch while you are translating.

For more information on the dictionary structure and on how to construct an SGML-based file, see:

- “Dictionary entry structure”
- “Creating an SGML-based dictionary”

Dictionary entry structure

A dictionary entry is divided into the following levels:

Entry

This level contains general or administrative information, such as the date, author, or origin of an entry, and the headword itself. This information applies to all other levels of the dictionary entry.

Homonym

This level contains grammatical and syntactic information, such as which part of speech a headword is or how it is to be abbreviated and hyphenated.

Sense

This level contains semantic variations of a headword, such as different areas of meaning and usage.

Target

This level contains all information applying to one translation variant of a headword, such as definition or usage. For example, if a headword can be translated in two different ways, there is target level information for both translation variants.

The entry level can have any number of homonym levels. For example, there are separate homonym levels for the noun **copy** and the verb **copy**.

A homonym level can also have any number of sense levels depending on the individual definition of a headword or contextual information. For example, the noun **bank** can mean both the river bank or a financial institution.

A sense level can have any number of target levels depending on the individual translation variants. For example, the German word **Schnecke** has two English translations, **snail** and **slug**. The entry fields and how they are distributed over the four levels is only suggested by **OpenTM2**. The entry fields **Headword** and **Translation** are mandatory. All other predefined fields can be deleted or renamed. You can even omit particular levels. For example, you can omit the homonym and sense level if you want to have only headwords and translations in your dictionary.

You can also add new fields to any of the levels. You can do this when creating a new dictionary in the "New dictionary" window by selecting **Add fields...** or by including them in the SGML dictionary file.

For the list of all predefined entry fields and their corresponding SGML tags see [List of dictionary-entry fields and their SGML tags](#).

Dictionary search criteria

To search for one or more dictionary entries, the specified term can be used as one of the following search criteria:

Headword

The dictionary entry for the specified term is retrieved.

Synonyms

All entries that have the specified term in the synonym field of the dictionary entry are retrieved.

Related terms

All entries that have the specified term in the related terms field of the dictionary entry are retrieved.

Abbreviations

All entries that have the specified term in the abbreviation field of the dictionary entry are retrieved.

Neighborhood

If the specified term exists, it is listed together with terms preceding and following it in alphabetical order. If not, the terms that would precede and follow it are listed.

Compounds

All compounds that begin with the specified term are retrieved and displayed as a list. For example, if you search for the term **data**, the possible compounds that would be retrieved are **data processing**, **data model**, and **data processing division**.

The first four entry fields can only be used as search criteria when they exist at the entry level shown in the following table. The entry fields are:

Entry field	Level
Headword	Entry
Synonym	Sense
Other Related Terms	Sense
Abbrev./Fullform	Sense

In the **OpenTM2** default entry structure, these fields are defined at these levels.

If you are importing an SGML dictionary file and want to use any of the above search criteria, see [List of dictionary-entry fields and their SGML tags](#) for the SGML tags required.

Overview and terminology

Dictionaries are used during analysis and translation:

- During the analysis of a document one or more dictionaries can be searched. This is useful to find *new terms* (terms in the document that are not in the selected dictionaries) or to find *found terms* (terms in the document that are already in the referenced dictionaries). The new terms can be added to the dictionary to be used during translation. The found terms can be used to fill a dictionary with entries extracted from other dictionaries.
- During the translation of a document, the "Dictionary" window displays either all translations that are found in the folder dictionaries for the terms in the current segment or only the first translation found. This depends on whether you marked the box on the "Display" page of the "Customize Translation Functions" window.

The *properties* of a dictionary are its main characteristics such as the source language, the dictionary structure, information on its location, and a brief description of its contents. You can specify them in the "Dictionary Properties" window.[dictionaryproperties](#)

Parts of these properties can also be displayed as *details* (or *view details*) in the "Dictionary List" window.

What you can do with dictionaries

OpenTM2 offers several lookup functions for terms in dictionaries and their data:

- You can search for terms using general search criteria and filters defined by yourself (Look up a Term function).
- You can display all data of a dictionary entry or parts of it (Lookup Entry function).
- You can add and change dictionary entries (Edit an Entry function).

OpenTM2 dictionaries can be printed to a printer or to a file. You can select the entry fields to be printed and how to arrange them in the printout. This layout is defined in a print format file. You can also use filters to print selected entries only.

A dictionary can be shared with other users by means of the dictionary *export* function, or if it has been defined to reside on a shared disk when it was created. [dictionarysharing sharingdictionaries](#)
If you receive an exported dictionary, you can *import* it into your system.

During import, you can also *merge* the contents of one dictionary into another. In addition, you can create dictionaries tailored to suit your own needs. [dictionarypersonal](#) To create a new dictionary, you must determine the *dictionary structure*, this means, which data elements in a dictionary are important to you. You can start using the **OpenTM2** default structure or using other existing dictionary structures that you can subsequently change.

OpenTM2 dictionaries can be protected with a password against unauthorized changes if necessary. Most of the processing functions **OpenTM2** offers for dictionaries can be started from the "Dictionary List" window.

Prerequisites

The dictionary must exist (except if you import a dictionary).

Calling sequence

Select:

1. The "Dictionary List" window in the **OpenTM2** main window.
2. A dictionary with which you want to work. Skip this step when importing or creating.
3. An action from the **File** menu.

The "Dictionary List" window is displayed.

This window displays a list of the existing dictionaries. When you select a command from one of the menus, you are taken to another window where you can specify processing options.

Creating an SGML-based dictionary

An SGML-based file contains dictionary data and SGML tags describing the data structure and the relation between the data elements.

SGML-based dictionaries can be created by:

- The **OpenTM2** dictionary export function
- Data conversion programs or database macros
- Editing a file (only recommended for small data volumes)

If you have dictionary data in a format other than SGML, you must convert it before you can import it.

Importing the SGML-based file into **OpenTM2** (which can be either ASCII, Ansi or Unicode) converts the dictionary into the internal format recognized by **OpenTM2**.

The structure of an SGML-based dictionary

An SGML-based dictionary is divided into the following:

1. A header section containing general dictionary information such as source language, target language(s), and creation date.
2. A mappable section specifying the structure of a dictionary entry. That is, all the entry fields that make up a dictionary entry and the relation between these entry fields.
3. An entry section with dictionary entries appearing one after another.

The dictionary is enclosed by a dictionary start tag <dict> and a dictionary end tag </dict>.

Header section

The header section contains general information about the dictionary such as:

- Type (for example, bilingual)
- Source language (for example, English (U.S.))
- Date when it was created
- Codepage encoding

The dictionary header is marked with a header start tag <header> and a header end tag </header>.

The only information that you must provide in the header section is the source language of the dictionary. You can specify any of the languages for which you have a language support file installed (if you need to specify another language, see [What you can do for other languages](#)).

OpenTM2 needs this information for reducing terms to their stem form when looking them up in the dictionary.

The header section can have the following tags:

Start tag	End tag	Description
<type>	</type>	Type of dictionary
<source>	</source>	Source language
<ltarget>	</ltarget>	Target language
<descript>	</descript>	Description of the dictionary, up to 255 characters long.
<createdate>	</createdate>	Date when dictionary was created
<CodePage>	</CodePage>	SGML-Format, can be UTF16, or ASCII=cp, or ANsi=cp

Example of a header section

```
<header>
<type>Bilingual</type>
<source>English (U.S.)</source>
<ltarget>German (national)</ltarget>
<descript>Basic Dictionary - English -> German</descript>
<CodePage>ASCII=850</CodePage>
<createdate>31/05/94</createdate>
</header>
```

Maptable section

The maptable section determines how a dictionary entry is structured. It serves as a model for the structure of an entry. The fields you defined in the maptable section are reflected in the dictionary entries.

The maptable contains the total of all allowed fields (up to a maximum of 38) in a dictionary entry. If a dictionary entry contains entry field tags not included in the maptable section, the information between the tags is not imported during dictionary import and only the entry fields listed in the maptable are taken into account. The information between a start tag and its end tag contains the name you want to give the entry field of the dictionary. You find these names displayed when you look up or edit a term in a dictionary.

For example, if you specify <hdterm> Source Term </hdterm>, the terms you look up in the dictionary are shown under *Source Term*. If you want to use another name, rename *Source Term* into any other name you would like to use instead.

The following is an example of a maptable section:

Example of a maptable section

```
<maptable>
<hdterm>Source term</hdterm>
<hom>
<epos>English category</epos>
<sense>
<edef>Formal English definition</edef>
<target>
<trans>NL term</trans>
</target>
</sense>
</hom>
</maptable>
```

The maptable begins with the start maptable tag <maptable> and ends with the end maptable tag </maptable>. The <hom>, <sense>, and <target> start tags with their respective end tags mark the homonym, sense, and target levels.

If you want to use the search criteria described in [Searching for a dictionary entry](#), use the following start and end SGML tags:

Searching for ...	Level	Tags
Headwords	Entry	<hdterm> and </hdterm>
Abbreviations	Sense	<eabbr> and </eabbr>
Synonyms	Sense	<esyn>

Searching for ...	Level	Tags
		and </esyn>
Related terms	Sense	<erel> and </erel>

If you use the tags listed above, **OpenTM2** establishes a correct relation between the entry field name and the term you want to look up in the dictionary.

If you want **OpenTM2** to add date information automatically to your dictionary entries, add the Creation Date and Update fields to the maptable.

If you want OpenTM2 to show style indicators in the dictionary lookup window, add the Style and the Trans Style fields to the maptable.

You can add any number of individually defined entry fields to any level. The entry field names are numbered and automatically mapped to an entry field name denoted by you in the maptable. Entry fields you add to the entry, homonym, or sense level start with the tag `<euser id=xxx>` and end with the tag `</euser>`, where `xxx` is the running number. Entry fields you add to the target level start with `<tuser id=xxx>` and end with `</tuser>`.

You can control the display panel (the panel on which a field is displayed in the "Lookup Entry" window) using the `displevel`= attribute. Use `displevel=1` for fields to be displayed in panel 1, `displevel=2` for fields to be displayed on panel 1 and 2 and `displevel=3` for fields to be displayed on panel 1, panel 2, and panel 3. To omit fields from the display in the "Lookup Entry" window use `displevel=0`.

The size of a dictionary field in the "Edit Entry in Dictionary" window is controlled using the `entrytype` attribute. Specify `entrytype=1` for single line fields and `entrytype=2` for multi-line fields.

In order to display the contents of a dictionary field as additional dictionary data in the dictionary proposal area of the TranslationEnvironment specify the `autlookup` attribute for the field.

Example of date-information fields and user-defined fields

```

<maptable>
<hdterm>...</hdterm>
.
.
<ecrdate>Creation date</ecrdate>
.
.
<hom>
.
.
<sense>
.
.
<euser id=1 displevel=2 entrytype=1 autlookup>Entry code</euser>
<elupdate displevel=3 entrytype=1>Last update</elupdate>
.
.
<target>
<trans>...</trans>
<tcrdate displevel=3 entrytype=1>Translation creation date</tcrdate>
<tlupdate displevel=3 entrytype=1>Translation update</tlupdate>
<tuser id=2 displevel=3 entrytype=2>Related definition</tuser>
</target>
</sense>
</hom>
</maptable>

```

Entry section

The entry section lists the actual entries in the dictionary one after the other. Each entry starts with an `<entry>` tag and ends with an `</entry>` tag. Each entry contains the same entry field tags as used in the maptable section. Whereas the maptable contains the general description of the structure of an entry, the entry section contains real data.

The levels below the entry level can be repeated as often as necessary, for example, to contain multiple senses or translations.

For an example of an SGML-based dictionary see [Sample of an SGML-based dictionary](#).

List of dictionary-entry fields and their SGML tags

The following table shows a list of all dictionary-entry field tags as they are predefined in the -Master- model dictionary.

Level	Entry field name	Start tag	End tag
Entry	Headword	<hdterm>	</hdterm>
Entry	Reliability Code	<erlcode>	</erlcode>
Entry	Author	<eaauthor>	</eaauthor>
Entry	Creation Date	<ecrdate>	</ecrdate>
Entry	Status Code	<estatcode>	</estatcode>
Homonym	Part of Speech	<epos>	</epos>
Homonym	Morphology	<emorph>	</emorph>
Homonym	Hyphenation	<ehyph>	</ehyph>
Sense	Source of Headword	<esrc>	</esrc>
Sense	Abbrev./Fullform	<eabbr>	</eabbr>
Sense	Author of Update	<eauthupdate>	</eauthupdate>
Sense	Last Update	<elupdate>	</elupdate>
Sense	Definition	<edef>	</edef>
Sense	Source of Definition	<esdef>	</esdef>
Sense	Synonym	<esyn>	</esyn>
Sense	Other Related Terms	<erel>	</erel>
Sense	Context	<econtext>	</econtext>
Sense	Source of Context	<escontext>	</escontext>
Sense	Comments	<comment>	</comment>
Sense	Note on Usage	<eusage>	</eusage>
Sense	Idiom	<eidiom>	</eidiom>
Sense	Style	<estyle>	</estyle>
Target	Language	<tlanguage>	</tlanguage>
Target	Translation	<trans>	</trans>
Target	Company/Subject Code	<tsubjcode>	</tsubjcode>
Target	Source of Translation	<tsrc>	</tsrc>
Target	Abbrev./Fullform	<tabbr>	</tabbr>
Target	Reliability Code	<tricode>	</tricode>
Target	Author	<tauthor>	</tauthor>
Target	Author of Update	<tauthupdate>	</tauthupdate>
Target	Creation Date	<tcrdate>	</tcrdate>
Target	Last Update	<tlupdate>	</tlupdate>
Target	Status Code	<tstatcode>	</tstatcode>
Target	Part of Speech	<tpos>	</tpos>
Target	Morphology	<tmorph>	</tmorph>
Target	Hyphenation	<thyph>	</thyph>
Target	Definition	<tdef>	</tdef>
Target	Source of Definition	<tsdef>	</tsdef>
Target	Synonym	<tsyn>	</tsyn>
Target	Other Related Terms	<trel>	</trel>
Target	Context	<tcontext>	</tcontext>
Target	Source of Context	<tscontext>	</tscontext>
Target	Comments	<tcomment>	</tcomment>
Target	Note on Usage	<tusage>	</tusage>
Target	Idiom	<tidiom>	</tidiom>
Target	Trans Style	<tstyle>	</tstyle>

Sample of an SGML-based dictionary

The following is an example of an SGML-based dictionary containing a header section, a maptable, and two entries. The text to the right of the vertical line is explanatory information or comments to the tags. All explanatory information must be omitted when you actually create an SGML-based dictionary.

```

<dict>
  <header>
    <source>English (U.S.)</source>
    <descriptor>Basic Dictionary - English -> German</descriptor>
  </header>
  <mptable>
    <hdterm>Source term</hdterm>
    <eauthor>Created by</eauthor>
    <ecrdate>Term Creation Date</ecrdate>
    <hom>
      <epos>Category</epos>
      <sense>
        <edef>Definition</edef>
        <euser id=210>Private glossary</euser>           user-defined entry field
                                                       added to sense level
      <target>
        <tsubjcode>Subject code</tsubjcode>
        <trans>Translation</trans>
        <tuser id=1>Project code</tuser>
      </target>
    </sense>
    <hom>
  </mptable>
  <entry>
    <hdterm>file</hdterm>
    <eauthor>Fred Miller</eauthor>
    <ecrdate>940401</ecrdate>
    <hom>
      <epos>n</epos>
      <sense>
        <target>
          <tsubjcode>EDV</tsubjcode>
          <trans>Datei</trans>
        </target>
      </sense>
    <hom>
  </entry>
  <entry>
    <hdterm>abend</hdterm>
    <eauthor>Peter G.</eauthor>
    <ecrdate>940501</ecrdate>
    <hom>
      <epos>v</epos>
      <sense>
        <target>
          <tsubjcode>EDV</tsubjcode>
          <trans>abnormal beenden</trans>
          <tuser id=4>AS400</tuser>
        </target>
      </sense>
    <hom>
  </entry>
</dict>

```

start of dictionary
 start of header section
 source language (required)
 end of header section
 start of mptable section
 user-defined entry field
 added to sense level
 user-defined entry field
 added to target level
 end of mptable section
 start of first entry
 headword: file
 author: Fred Miller
 creation date: 4/01/94
 start of homonym level
 part of speech: noun
 start of sense level
 start of target level
 subject code
 translation
 end of target level
 end of sense level
 end of homonym level
 end of first entry
 start of second entry
 headword: abend
 author: Peter G.
 creation date: 5/01/94
 start of homonym level
 part of speech: verb
 start of sense level
 start of target level
 subject code
 translation
 user-defined entry field
 not in mptable, field
 is ignored when this
 dictionary is imported
 end of target level
 end of sense level
 end of homonym level
 start of homonym level
 part speech: noun
 start of sense level
 start of target level
 subject code
 translation
 end of target level
 end of sense level
 end of homonym level
 end of second entry
 end of dictionary

Creating a dictionary

OpenTM2 offers you several ways of setting up a new dictionary.

- If you do not have any existing terminology in machine-readable form, you must create a completely new dictionary. You do this by determining the dictionary properties. In particular, you must define a dictionary structure. You can use a default structure offered by **OpenTM2**, or you can use the structure of an existing dictionary in **OpenTM2** and change it. A newly created

dictionary is empty at first but you can add entries from a new terms list built during document analysis or at any stage during the translation process. In this way you can create dictionaries that contain specific terms related to your translation projects.

- During analysis, **OpenTM2** can generate a found terms list that contains all terms of the document that exist in the referenced dictionaries. **OpenTM2** can also copy the entry data of these terms into a separate dictionary.
- If you have a terminology file in a format of your own, you must generate an external SGML-based dictionary and you must import it into **OpenTM2**. In this case, a new dictionary is created with your terminology and the entry structure as defined in the SGML file is taken.

If you create a new dictionary via the "New Dictionary" window in **OpenTM2** and you do not use the modeling option, the following entry fields are offered as default fields:

dictionarydefault entry fields	Level
Entry field	
Headword *)	Entry
Part of Speech	Homonym
Abbrev./Fullform *)	Sense
Definition	Sense
Synonym *)	Sense
Other Related Terms *)	Sense
Context	Sense
Translation	Target
Company/Subject Code	Target

The entry fields marked with *) can be used as predefined search criteria in the "Look up a Term" window (see [Searching for a dictionary entry](#)).

If you are working with a more comprehensive structure and require more entry fields, select **-Master-** in the "Use Existing Dictionary as Model" window, which offers an extensive dictionary structure. You can rename or delete any fields from this set of entry fields and you can also add new user-defined fields to it. **OpenTM2** adds and updates time stamp information automatically, provided the following date fields are selected in the "New dictionary" window from the **-Master-** model dictionary.

Entry field	Level	Contents
Creation Date	entry	The date when a headword was added to a dictionary.
Last Update	sense	The date when information at the sense level of an entry was added or modified.
Creation Date	target	The date when a translation for a headword was added.
Last Update	target	The date when a translation entry field was last updated.

Prerequisites

None.

Calling sequence

Select:

- The "Dictionary List" window
- **New...** from the **File** menu

The "New Dictionary" window is displayed (see [Figure 101](#)).

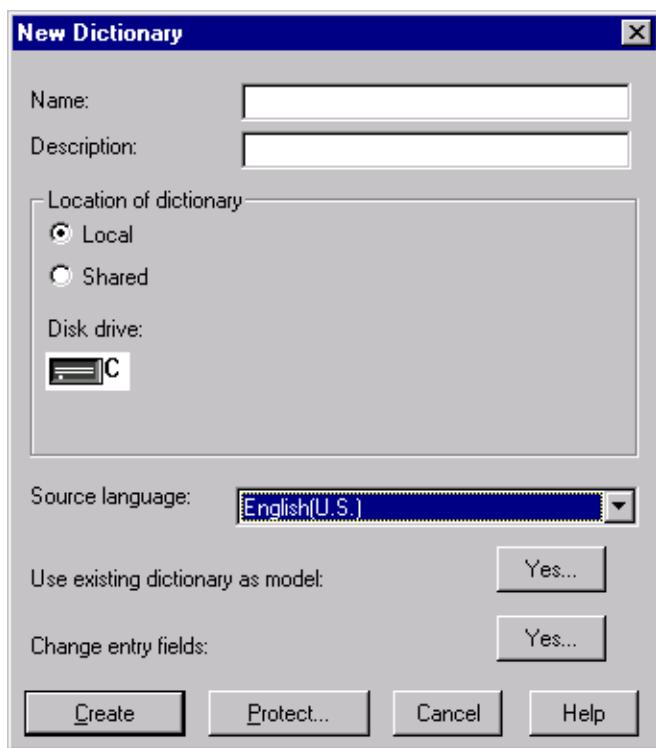


Figure 101. New Dictionary window

Options and parameters

Name

Type a name of your choice for the new dictionary. This name can be up to eight alphanumeric characters long.

Description

Type a description for the new dictionary. The description can be up to 255 alphanumeric characters long.

Location of dictionary

Specify where to place the new dictionary. It can be on a **local** disk drive (only you can use it) or on a **shared** disk drive (several translators can use it simultaneously). See [#unique_132](#) to learn about sharing dictionaries. Select the drive on which you want the new dictionary to reside. A dictionary grows with time, so select a drive with enough space.

Source Language

Select a source language from the list of installed languages displayed in the list box.

Use existing dictionary as model

If you do not want to determine the dictionary entry structure yourself, you can use the structure of an existing dictionary as a model by clicking **Yes....**. This takes you to the "Use Existing Dictionary as Model" window where you can select a dictionary as model. Click **Select** or **Cancel** to return to the "New Dictionary" window. For more information on this option see [Using an existing dictionary as model](#).

Change entry fields

If you want to change the dictionary entry structure (add, delete, or rename entry fields), click **Yes...**, which takes you to the "Change Entry Fields" window where you can define your changes. Click **Select** or **Cancel** to return to the "New Dictionary" window. For more information on this option see [Changing dictionary entry fields](#).

To avoid unauthorized modification of a dictionary, you can protect it by clicking **Protect...**, which takes you to the "Protect Dictionary" window where you can type a password of your own. Click **Protect** to return to the "New Dictionary" window. From now on, the dictionary can only be modified when the correct password is supplied. For more information on how to protect and unprotect dictionaries see [Protecting a dictionary](#).

Click **Create** to create the new dictionary.

Results

The dictionary is created but still empty. When the dictionary has been created, its entry structure is fixed and can no longer be changed. You can now start to fill it by importing an SGML-based file,

by entering terms from a new terms list, or by editing dictionary entries.

Using an existing dictionary as model

All dictionaries have a dictionary entry structure. In **OpenTM2** you can use the given default structure, use an existing dictionary structure with or without changes, or define a new structure based on all the available entry fields.

Prerequisites

Decide on which dictionary entry fields your dictionary is to contain.

Calling sequence

Select:

1. The "Dictionary List" window.
2. **New** from the **File** menu. The "New Dictionary" window is displayed.
3. **Yes...** next to the **Use existing dictionary as model** option.

The "Use Existing Dictionary as Model" window is displayed (see [Figure 102](#)).

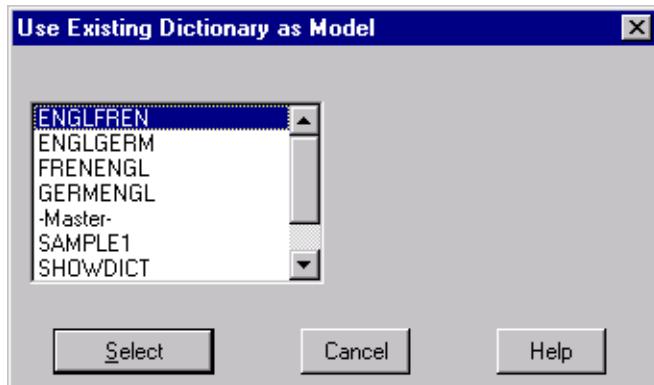


Figure 102. Use Existing Dictionary as Model window

Options and parameters

The list of existing dictionaries is displayed in the list box. Select a dictionary with the structure you want to use as a model from this list or select **-Master-**, which contains all the entry fields predefined by **OpenTM2**.

Click **Select** to return to the "New Dictionary" window where you can view and change the selected entry structure by selecting the **Change entry fields** option.

Results

You have determined which entry structure to use as a base for the new dictionary.

Changing dictionary entry fields

dictionarychanging entry fields Select this function to make modifications to a dictionary entry structure. You can delete entry fields, rename entry fields, or add new user-defined entry fields. You can determine the size of the entry field and what entry field data you want to have displayed on which panel of the "Lookup Entry" window.

Prerequisites

Decide on which dictionary entry fields your dictionary is to contain.

Calling sequence

Select:

1. The "Dictionary List" window.
2. **New...** from the **File** menu. The "New Dictionary" window is displayed.
3. **Yes...** next to the **Change entry fields** option.

The "Change Entry Fields" window is displayed (see [Figure 103](#)).

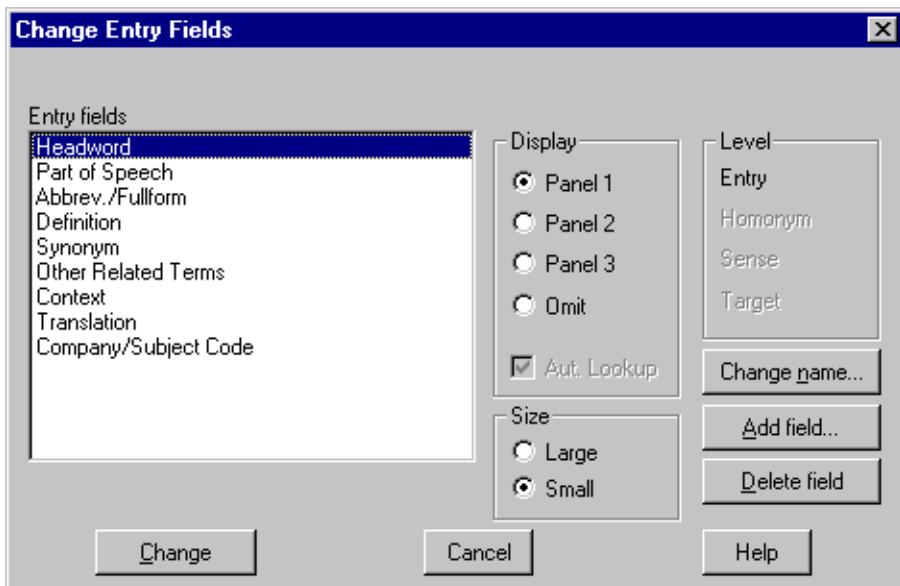


Figure 103. Change Entry Fields window

Options and parameters

Entry fields

This list box contains all the entry fields defined for this dictionary. Select one entry field at a time and define your settings for this field.

Display

If you want this entry field to be displayed in one of the panels of the "Lookup Entry" window, decide on which one. This means choose between **Panel 1** (for the most relevant data), **Panel 2**, and **Panel 3**.

Select **Omit** if you want the entry field and its data not to be displayed on any of the lookup panels. The entry field is not deleted from the entry structure. Select **Aut. Lookup** to display the complete entry field information in the window showing the found dictionary proposals during translation.

Level

To specify the level you want the entry field to belong to, select a corresponding entry field so that the required level is active. It can be **Entry**, **Homonym**, **Sense**, or **Target**.

Size

Select one of the two choices:

Small

Is adequate for an entry field of about a line in length (the maximum is 255 characters).

Large

Is needed for an entry field consisting of several lines of data. Select this size only if absolutely necessary because the use of too many large fields decreases the performance of **OpenTM2**.

For each selected entry field you can click one of the following:

Change Name...

To rename the entry field. See [Renaming a dictionary entry field](#) for details.

Add Field...

To add a new user-defined entry field to the list of entry fields. For more detail see [Adding a user-defined entry field](#).

Delete field

To remove the entry field from the entry structure.

When all of the entry field settings are complete, click **Change**. This takes you back to the "New Dictionary" window.

Results

The entry structure of the selected dictionary is changed according to your specifications, this means the dictionary can have more or fewer entry fields, changed entry field names and sizes, and changed lookup display options.

Renaming a dictionary entry field

dictionaryrenaming entry fields In **OpenTM2**, all entry fields are automatically assigned a name. You can change the field name to any other name. The changed names are used in all windows where dictionary entry data is displayed. For example, you may want to change *Headword* to *Term*, or *Part of speech* to *Category*. You can also change the field names to another language of your preference.

You can change the entry field names of an existing dictionary when you create a new dictionary.

Prerequisites

Decide which dictionary entry field names you want to use in your dictionary.

Calling sequence

When creating a new dictionary, select:

1. The "Dictionary List" window.
2. **New...** from the **File** menu. The "New Dictionary" window is displayed.
3. **Yes...** next to the **Change entry fields** option.
4. The entry field you want to rename in the "Change Entry Fields" window.

When changing the properties of an existing dictionary, select:

1. The "Dictionary List" window
2. The dictionary to which you want to make changes
3. **Properties...** from the **File** menu
4. The entry field you want to rename in the "Dictionary Properties" window

The "Change Entry Field Name" window is displayed.

Options and parameters

Type the new field name in the **Displayed as** field. Click **Change** to rename the selected field.

Results

The selected dictionary entry field is renamed. You are returned to the "Change Entry Fields" window when you create a new dictionary or to the "Dictionary Properties" window when you change the properties of an existing dictionary.

Adding a user-defined entry field

dictionaryadding your own entry fields The predefined entry fields in **OpenTM2** dictionaries are only examples of possible fields you can use when you create a new dictionary and determine its structure.

You can add your own user-defined fields to any of the four levels that make up the dictionary structure.

You can only add user-defined fields when you create a new dictionary. When the dictionary already exists, you cannot change its structure.

Prerequisites

Decide which dictionary entry fields you want to add to your dictionary structure.

Calling sequence

The "Dictionary List" window

1. **New...** from the **File** menu
2. **Yes...** next to the **Change entry fields** option in the "New Dictionary" window
3. An entry field belonging to the level to which you want to add a field

The "Add Entry Field" window is displayed.

Options and parameters

Type a name for the entry field you want to add in the **Displayed as** field. Click **Add** to add the new field to your structure.

Results

The entry field is added to the dictionary structure at the same level as the entry field you have selected before clicking **Add field....** You are returned to the "Change Entry Fields" window.

Defining a dictionary filter

dictionary filter filterdefinition of A dictionary filter is a means by which data in a dictionary can be selected so that only entries that meet specified conditions are displayed or printed.

In **OpenTM2** filters can be used in combination with printing (see [Printing a dictionary](#)) or dictionary lookup (see [Looking up a dictionary entry](#)). **filterfor printing dictionary extracts filterfor looking up dictionary entries** You can use an existing filter, change an existing one, or create a new one.

In **OpenTM2** a filter condition consists of:**filterconditions**

- An entry field (such as **Headword**, **Translation**)
- An operator (such as = or >)
- An operand (such as a*)

For example, the condition Headword LIKE = 'a*' selects all (and only) the entries beginning with the letter a.

There are two types of operators:

1. (=, <>, <, >, <=, >=, LIKE, BETWEEN and IN form **expressions** (for example, Headword = 'Test').
2. AND, OR, (,) and **NOT**) join valid expressions to form a larger expression (for example, Headword = 'Test' AND Translation = 'Test').

The operators have the following meanings:

=

The field must contain exactly the value specified.

<>

The field must not contain the value specified.

<

The field can contain only values that are less than the value specified, for example, preceding the specified value in alphabetical sequence.

>

The field can contain only values that are greater than the value specified, for example, following the specified value in alphabetical sequence.

<=

The field can contain only values that are less than or equal to the value specified.

>=

The field can contain only values that are greater than or equal to the value specified.

LIKE

The field can contain values that are similar to the value specified, for example, they start with the specified value.

BETWEEN

The field can contain values that are in a range between one value and another.

IN

The field can contain values that match up to six different values.

AND

This operator joins two expressions so that both expressions must match in order to let the entry pass the filter.

OR

This operator joins two expressions so that one or both expressions must match in order to let the entry pass the filter.

NOT

This operator specifies that the filter let pass only those entries that do not match the specified value.

(and)

These operators are used to group expressions. Expressions in parentheses are processed before expressions without parentheses.

Note:

All filters depend on the dictionary entry structure. This means that a filter created for one dictionary possibly refers to entry fields that are not contained in another dictionary. In such cases, it is recommended to create a new filter for each dictionary.

Prerequisites

Either the "Look up a Term" window or the "Print Dictionary" window is active.

Calling sequence

Select the **Use filter** option.

1. Click **Edit...**

The "Edit Dictionary Filter" window is displayed (see [Figure 104](#)).

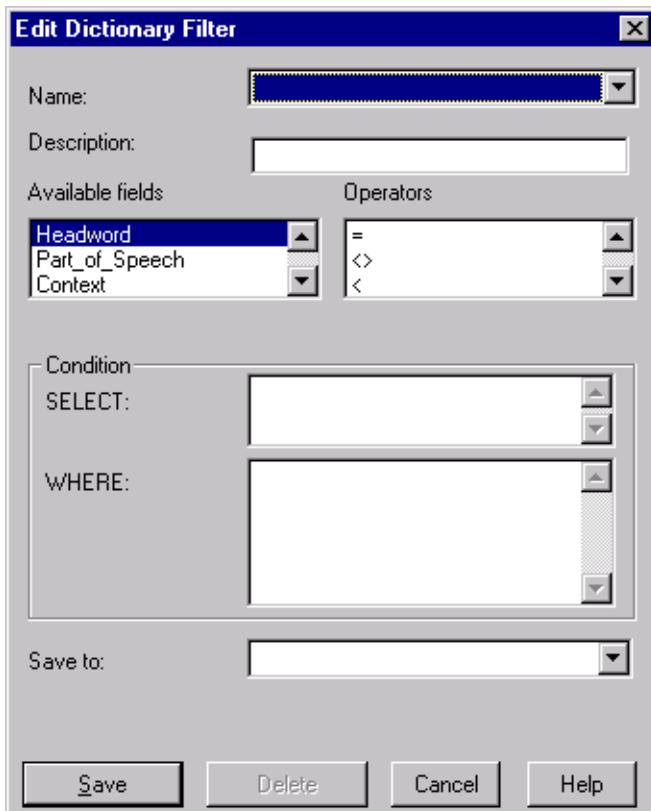


Figure 104. Edit Dictionary Filter window

Options and parameters

The dictionary-filter parameters are as follows:

Name

Lists the name of all available filters. If you specified or selected an existing filter in the **Print Dictionary** or "Look up a Term" window, the name of this filter is displayed first in the list. If you specified a new filter in either of these windows, this field is empty and the new name is only displayed in the **Save to** box. If you want to change an existing filter, select this filter. If you want to create a new filter, skip this box.

Description

Displays the description of a filter if one was entered. For a new filter, type a descriptive text.

Available fields

Lists the name of all entry fields that a dictionary entry consists of. If you want all entry fields to appear in the printout or in the "Lookup Entry" window (depending on which window you came from) but do not specify any conditions for the entry fields, you can skip this box. If you want specific entry fields to appear in the printout or in the "Lookup Entry" window but do not want to specify any conditions for them, double-click these entry fields. They will appear in the **SELECT** box. If you also want to specify conditions for one or more entry fields using one of the operators =, <>, <, >, <=, >=, **LIKE**, **BETWEEN**, and **IN**, double-click the entry field and then double-click the operator. The "Set Dictionary Condition" window is displayed in which you can specify one or more values, depending on the operator that you selected. The condition that you specify is displayed in the **WHERE** box after you leave the "Set Dictionary Condition" window.

Note:

The operators AND, OR, NOT, (, and) can only be selected for the WHERE box. Lists the entry

fields defined for the selected dictionary.

Operators

Lists the available operators that can be used with an entry field to specify a filter condition.

Condition

Displays the defined filter condition:

SELECT

Lists the entry fields that you selected in the **Available fields** box. Only those listed here will appear in the printout or in the "Lookup Entry" window, depending on which window you came from. To add more entry fields, double-click them in the **Available fields** box. To include all fields, leave this box blank or type an asterisk (*).

WHERE

Contains the condition that you specified and that a dictionary entry has to match in order to pass the filter. A condition expression consists of an entry field, an operator, and a value. To add an expression, double-click an entry field in the **Available fields** box and then double-click one of the operators =, <>, <, >, <=, >=, **LIKE**, **BETWEEN**, and **IN**. The "Set Dictionary Condition" window is displayed in which you can specify one or more values, depending on the operator that you select. The expression that you specify is displayed in the **WHERE** box after leaving the "Set Dictionary Condition" window. To join expressions, move the cursor between the expressions to be joined and then double-click the operator **AND**, **OR**, or **NOT**. For example, Headword = 'Test' AND Translation = 'Test'. To group expressions, move the cursor to the beginning or end of an expression and then double-click the operator (or).

Save to

Displays the filter name that you specified or selected in the **Print Dictionary** or "Look up a Term" window or selected from the **Name** list. You can type a new name or select the name of an existing filter. In the latter case, the existing filter is overwritten.

Select **Save** to create the new or changed filter. Select **Delete** to delete the displayed filter. For example, if you want to create a filter that selects only those entries in the dictionary that begin with the letter *a*, define the filter as follows:

1. Click the field name *Headword* in the **Available fields** list box to mark it.
2. Double-click the **LIKE** operator in the **Operators** list box. The "Set Dictionary Condition" window appears.
3. Type *a** or *A** at the cursor position and click **Set**. The input can be in uppercase, lowercase, or mixed-case characters.
4. The condition is displayed in the **WHERE** field.
5. Enter a name for the filter (for example, **ONLYA**) in the **Save to** combination box.
6. Click **Save** to create the filter.

Results

The changed or new filter is saved and can be used for dictionary lookup or printing. You are returned to the previous window.

Renaming a dictionary

dictionaryrenaming renamingdictionaries **OpenTM2** enables you to give a dictionary a new name.

Prerequisites

The dictionary must exist.

Calling sequence

Select:

1. The dictionary from the "Dictionary List" window
2. **Rename** from the **File** menu

Options and parameters

Rename to

Type a new name for the dictionary.

Adjust all references automatically

If you select this option, all references to the renamed dictionary are also changed.

Results

If you select **Rename**, the dictionary and, optionally, any references to it are renamed.

Deleting a dictionary

dictionarydeleting deletingdictionaries Deleting a dictionary can be useful when you have exported it or archived it outside of **OpenTM2** and no longer need it, and you require free space on your disk. For deleting a shared dictionary, see [Deleting a shared dictionary](#).

Prerequisites

Make sure that you have a backup copy of the dictionary before deleting it if it contains important data.

Calling sequence

Select:

1. "Dictionary List" window
2. The dictionary you want to delete
3. **Delete** from the **File** menu

Options and parameters

A confirmation message appears before the dictionary is deleted. If you select **No**, the delete function is canceled. If you select **Yes**, the dictionary is deleted.

Results

If you select **Yes**, the dictionary is deleted. References to this dictionary may still exist in certain folders. Update the folder properties so that the deleted dictionary is not referenced anymore.

Editing a dictionary entry

dictionaryupdating an entry dictionaryadding an entry dictionaryremoving an entry Individual dictionary entries can be changed, for example, a new translation can be added or an existing obsolete translation deleted.

You can edit a dictionary entry at any time before, after, or during translation.

Prerequisites

The dictionary must exist.

Calling sequence

There are several ways to get to the "Edit Entry in Dictionary" window:

- To add, change, or delete a dictionary entry outside the **Translation Environment**, select:
 1. The "Dictionary List" window
 2. The dictionary you want to open
 3. **Open** from the **File** menu
 4. The term to enter in the "Look up a Term" window
 5. **Headword** search type and click **Edit...**
- To add, change, or delete a dictionary entry in the **Translation Environment**, select:
 1. The document for translation in the "Document List" window
 2. **Open** from the **File** menu
 3. The term in the document to add or modify by marking it
 4. **Edit a term...** in the **Translate** menu
- Or select:
 1. The term in the "Dictionary" window and double-click it
 2. **Edit** in the "Lookup Entry" window

In all cases, the "Edit Entry in Dictionary" window is displayed (see [Figure 1](#)).

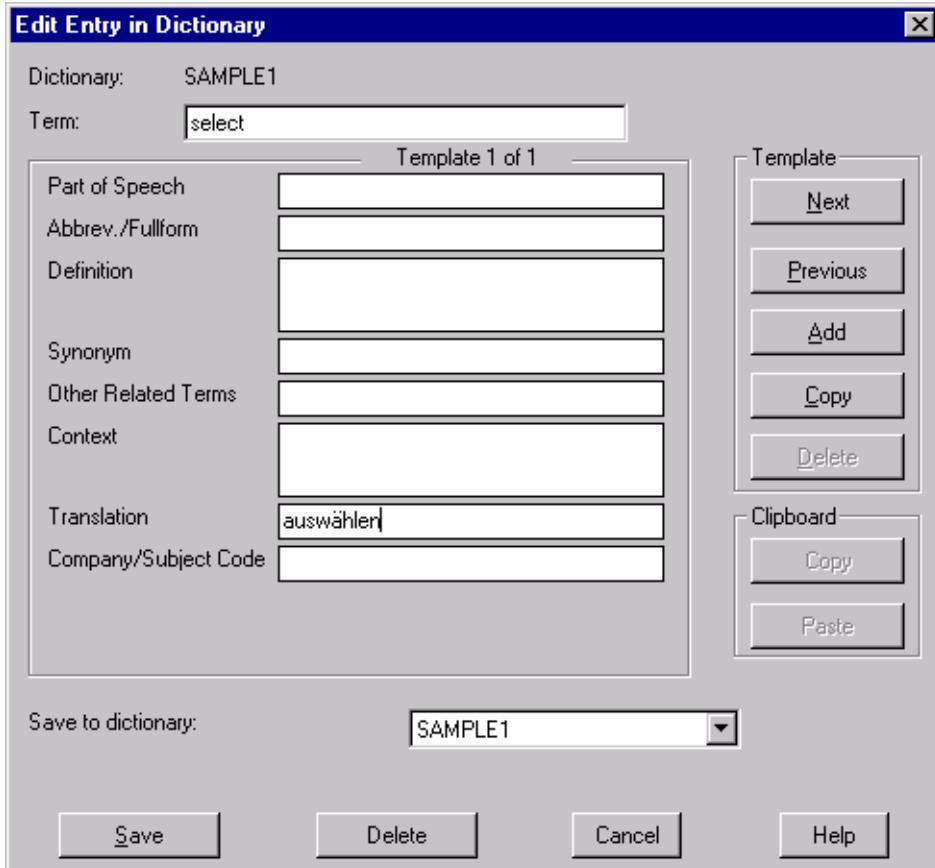


Figure 105. Edit Entry in Dictionary window

Options and parameters

The entry displayed in the "Edit Entry in Dictionary" window is split up into templates. A template is the complete dictionary entry information related to one translation. This window shows one template at a time.

If the term does not exist, only one empty template is displayed. If, on the other hand, the term exists, the first template (translation) is displayed.

Template 1 of 1

The term in the sample window has one translation. This box displays all data related to this translation, this means, each entry field with its name and its contents.

Template

The following options let you scroll through all templates of an entry or change complete templates.

Select:

Next

To select the following template.

Previous

To select the previous template.

Add

To add an empty template for a new translation variant.

Copy

To copy the contents of the currently displayed template into a new one where you can type over the contents to add the new translation. You would use **Copy** instead of **Add** in cases where only minor changes need to be made to a translation variant.

Delete

To delete the currently displayed template.

Clipboard

By selecting options in this box, you can move data from the current template to the clipboard and vice versa, for example, you can insert context information from the document in the **Translation Environment**.

Copy

To copy a marked text to the clipboard.

Paste

To copy a previously saved text from the clipboard to the current cursor position.

To delete the entire entry from the dictionary, click **Delete** at the bottom of the window.

Click **Save** to save your changes to the dictionary displayed in the **Save to dictionary** list box. In the **Translation Environment**, you can choose between any of the dictionaries opened for translation (defined as a folder property). Outside the **Translation Environment**, you can only save the data to the dictionary you explicitly selected.

Results

A new term is added to the dictionary, or an existing term is changed or deleted.

Exporting a dictionary

dictionaryexporting exportingdictionaries There are two ways to export dictionary data, within a folder in the internal **OpenTM2** format or as an external dictionary in SGML format.

Export a dictionary if you want to pass the dictionary on to other **OpenTM2** users or to create a backup copy of a dictionary. Protected dictionaries must be exported in a folder. For more information on folder export, see [Exporting a folder](#).

Exporting a dictionary into an SGML-based file is required if you want to use the dictionary data in environments other than **OpenTM2**. **dictionaryUnicode enablement**

Exporting a dictionary can also be part of the conversion of a dictionary into Unicode. To convert a dictionary into Unicode, you must first export it from OpenTM2 and then import it again. In both cases, you select **SGML Unicode** as export or import format.

Prerequisites

Sufficient disk space for the exported dictionary is available.

Calling sequence

Select:

1. The "Dictionary List" window
2. The dictionary you want to export
3. **Export...** from the **File** menu

The "Export dictionary" window is displayed (see [Figure 106](#)).

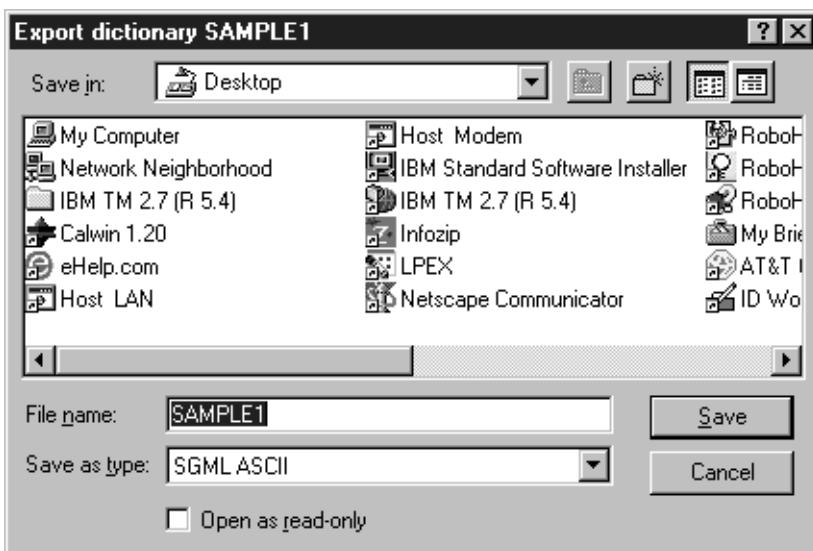


Figure 106. Export Dictionary window

Options and parameters

Save in

Select the drive and directory to which the dictionary is to be exported.

File name

Contains the name of the dictionary to be exported. You can overwrite it with a new name.

Save as type

Select the format in which the dictionary is to be stored. Select **SGML Unicode** if the export is part of your conversion of the dictionary into Unicode.

Click **Save** to begin exporting the dictionary. If you specified a new directory name, you are prompted to confirm whether you want to create this new directory.

Results

An SGML-based dictionary file is created at the specified location and can be given to other users or programs for further usage.

Importing a dictionary

dictionaryimporting importingdictionaries dictionarymerging when importing You can import a dictionary as an external dictionary (SGML-based) or within a folder in the **OpenTM2** format.

If you want to work with terminology created outside **OpenTM2**, create an SGML-based file (for more details, see [Creating an SGML-based dictionary](#)) and import this file into **OpenTM2**. You can either import the file into a new dictionary or merge the data into an existing dictionary.

Dictionaries imported in a folder are in **OpenTM2** format. If the dictionary does not exist, a new dictionary is created; if it does exist, the folder dictionary is merged into the existing one. For more information on folder import, see [Importing a folder](#).

When merging two dictionaries, the dictionary structure of the destination dictionary is maintained. Data in entry fields that exist only in the source dictionary and not in the destination dictionary will not be regarded and data may be lost. Entry fields not contained in the external dictionary will be empty in the dictionary after importing.

If you select to import an external dictionary under a new dictionary name, the "New Dictionary" window is displayed where all the properties are shown as they are coded in the SGML-based file. You can change certain properties if you want to, although this is not always advisable as the changed properties may be different from what is specified in the SGML file. **dictionaryUnicode enablement**

Importing a dictionary can also be part of the conversion of a dictionary into Unicode. To convert a dictionary into Unicode, you must first export it from OpenTM2 and then import it again. In both cases, you select **SGML Unicode** as export or import format.

Prerequisites

The SGML-based file for import must have been created according to the description in [Creating an SGML-based dictionary](#).

Calling sequence

Select:

1. The "Dictionary List" window
2. A dictionary if you want to merge dictionaries
3. **Import...** from the **File** menu

The "Import Dictionary" window is displayed (see [Figure 107](#)).

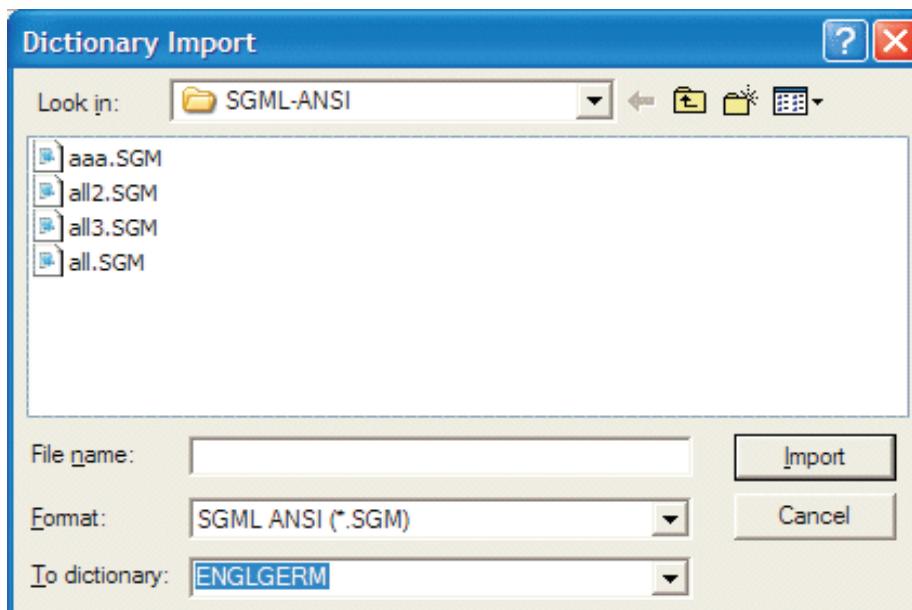


Figure 107. Import Dictionary window
[\[PIC\]](#)Figure shows Import Dictionary window

Options and parameters

Look in

Select the directory where the dictionary to be imported resides.

File name

Enter the name of the external dictionary to be imported.

Format

Select the format of the dictionary to be imported.

To dictionary

Contains the name of the selected dictionary to be imported. You can overwrite this name to create a new dictionary or select a name from the list box to merge your file into an existing dictionary.

Click **Import** to begin the import function.

If **OpenTM2** detects an incorrect SGML tag in the SGML-based file, a message is displayed, asking you if you want to skip the entry containing the error or to cancel the import process.

If you selected to import an external dictionary under a new dictionary name, the "New Dictionary" window is displayed where all the properties are shown as they are coded in the SGML-based file. For more information, see [Creating a dictionary](#). If you selected to merge the external dictionary into an existing dictionary in **OpenTM2**, remember that the dictionary structures may differ and that there may be identical terms. In case of conflicts you are prompted to specify additional options for merging dictionaries in the "Merge Dictionary Entries" window. For more information, see [Merging dictionaries](#).

Results

The external dictionary data is imported either under a new name or merged with an existing dictionary. You can begin to use it in **OpenTM2**. The external dictionary is unchanged.

Looking up a dictionary entry

dictionary[looking up an entry](#) There are several ways to look up dictionary entries in **OpenTM2**:

- During translation, **OpenTM2** automatically displays the translation of all terms in the currently active segment for which entries are found in the dictionaries selected for search in the folder. The dictionaries are searched in the order defined in the folder properties (see "[Changing the properties of a folder](#)"), and the first entry found is displayed in the dictionary window.
- To view the complete dictionary entry or to check if there are any further entries in other dictionaries, you can double-click the term, and the "Lookup Entry" window for this term is displayed, which allows you to select the dictionary (if the entry occurs in more than one of the folder dictionaries) and the lookup panel you want to see.
- You can also mark a term or a multiword term in the "Translation" window and select **Look up a term...** from the **Translate** menu. If the term is not found in the dictionaries, the "Look up a

"Term" window is displayed. If the term is found in the dictionaries, the "Lookup Entry"window is displayed containing the data for this term.

Outside the **Translation Environment** you can look up entries only in one dictionary at a time. How to search for terms in a dictionary is described in [Searching for a dictionary entry](#). How to edit an existing entry in a dictionary after you have retrieved it is described in [Editing a dictionary entry](#).

Prerequisites

- The dictionary must exist.
- The term you want to look up must exist in the dictionary.

Calling sequence

Outside the **Translation Environment**, follow this procedure to look up a dictionary entry:

1. Select the "Dictionary List" window.
2. Double-click the dictionary you want to open. The "Look up a Term"window is displayed, where you can type the term, select your search type, and define a filter if you want to restrict your search. For more detail on defining a filter, see "[Defining a dictionary filter](#)".
3. Type the term you want to look up in the **Term** field.
4. Select the **Headword** search option.
5. If the term exists in the dictionary, click **Look up** in the "Look up a Term"window. Otherwise **OpenTM2** offers you to add the term to the dictionary (see also "[Editing a dictionary entry](#)").

During translation in the **Translation Environment**, there are several ways to look up a dictionary entry:

- Using a displayed term in the "Dictionary" window, double-click the term you want to look up. The "Lookup Entry"window is displayed (see [Figure 108](#)).
- Using a term in the "Translation" window:
 1. Mark the term, possibly a multiword term.
 2. Select **Look up a term...** from the **Translate** menu.

If the specified term does not exist in the selected dictionary, the "Look up a Term" window is displayed (see [Figure 111](#)) showing existing terms preceding and following the marked term in alphabetical sequence. For more information on searching, see [Searching for a dictionary entry](#). In all cases, if the specified term exists in the dictionary, the "Lookup Entry" window is displayed (see [Figure 108](#)).

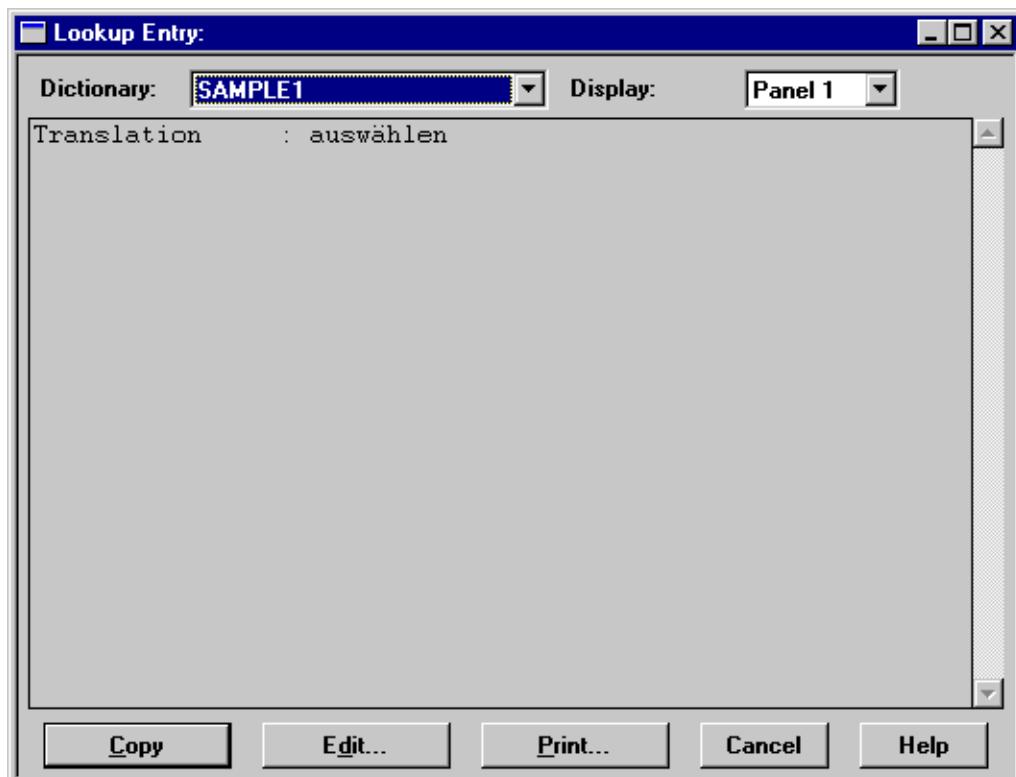


Figure 108. Lookup Entry window

Options and parameters

Dictionary

This field contains the name of the first dictionary where the term and its entry data are found. If the lookup function was called from the **Translation Environment** and the folder properties contain more than one dictionary, you can switch between the different dictionaries to see whether the same term is in more than one dictionary.

Display

In this field, you can switch between the lookup panels 1, 2, and 3 to display entry data of this term as defined in the dictionary structure.

The large center box contains entries for the found term in the displayed dictionary according to the panel selection.

Click **Print** to print the data of the term on your workstation printer.

Click **Edit...** if you want to edit the entry data of the found term in the "Edit Entry in Dictionary" window. For more information on editing, see [Editing a dictionary entry](#).

Click **Copy** to copy a marked text to the clipboard.

Click **Cancel** to leave the window.

Results

The entries for the selected term are displayed. The entries are changed, printed, or copied to the clipboard depending on the options you selected in this window.

Merging dictionaries

mergingdictionaries dictionarymerging Merging two dictionaries is only supported when you import an external dictionary into an existing one in **OpenTM2**. When you do this, remember that the dictionary structure of the destination dictionary has precedence. You are warned if the dictionary structures differ, and **OpenTM2** allows you to create a new dictionary instead, to prevent any loss of data.

If, during dictionary merge, the same term exists in both dictionaries, the "Merge Dictionary Entries" window is displayed (see [Figure 109](#)).

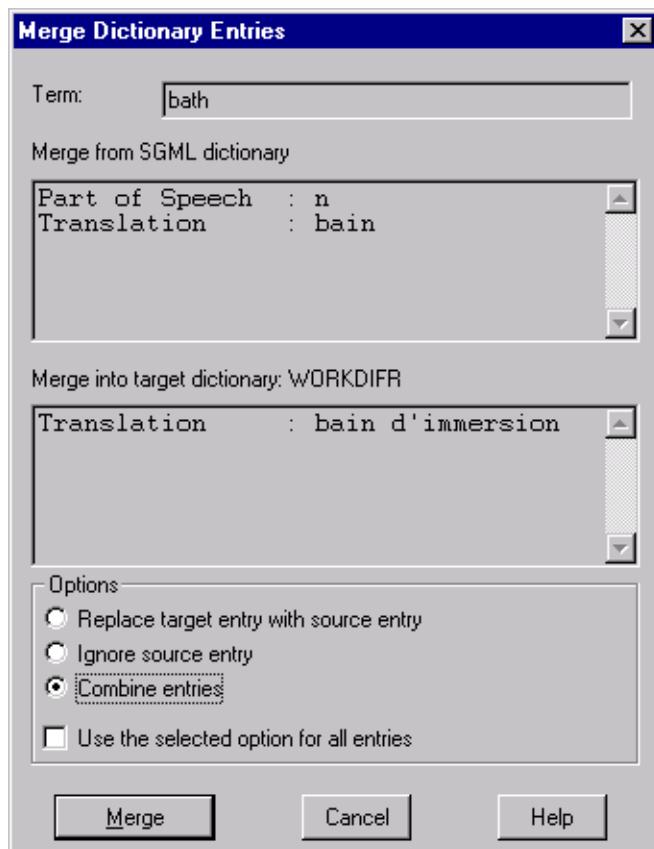


Figure 109. Merge Dictionary Entries window

Options and parameters

Term

Displays the term that exists in both dictionaries.

Merge from SGML dictionary

Contains the entries in the external dictionary for this term.

Merge into target dictionary

Contains the entries in the existing dictionary for this term.

Options

Depending on how you want to process the entries for the current term (displayed in the two boxes), select one of the following options or click **Cancel** to stop the merging:

Replace target entry with source entry

This option only takes the source dictionary entry into consideration. The destination dictionary entry is overwritten.

Ignore source entry

This option only takes the destination dictionary entry into consideration. The source dictionary entry is discarded.

Combine entries

This option takes both the source dictionary and the destination dictionary entries into consideration, combining them to form one entry. Provided that the dictionary structures do not differ, no data is discarded or overwritten.

Use the selected option for all entries

Select this option, if you do not want the "Merge Dictionary Entries" window to appear every time the same term is found in both dictionaries and you are sure that your selected merge option applies for all entries.

Click **Merge** to begin or to continue the merging.

Click **Cancel** to leave the merging and importing program.

Results

The external dictionary data is merged with the existing dictionary according to the specified options. The external dictionary is unchanged.

Opening a dictionary

To look at dictionary data, you must open it. In a dictionary, you can search for specified terms, you can look up the data of an existing term, you can change the entry data of an existing term, and you can add entries for new terms.

Opening a dictionary can be done in several ways outside the **Translation Environment** and during translation in the **Translation Environment**.

Opening a dictionary is automatically done, when you use one of the following functions:

- “Editing a dictionary entry”
- “Looking up a dictionary entry”
- “Searching for a dictionary entry”

Organizing a dictionary

dictionaryorganizing organizing dictionaries Organizing makes the dictionary compact and increases performance. Dictionaries to which you frequently make changes should be organized periodically.

If an error occurs during dictionary processing, you are prompted to organize the dictionary because organizing often rectifies the error.

Prerequisites

The dictionary must exist.

Calling sequence

Select:

1. The "Dictionary List" window
2. The dictionary you want to organize
3. **Organize** from the **File** menu

The organize procedure starts immediately.

Options and parameters

None.

Results

The dictionary is organized, informing you how many terms have been organized, and you can continue using it.

Printing a dictionary

dictionaryprinting printingdictionary extracts You can choose to print a complete dictionary or parts of it.

If you only want to print part of a dictionary, for example all entries with a headword starting with the letter a, or only some of the entry fields, or only the headwords and the translations, you can define a filter for the print function.

You must define the layout of your print output in format files that you edit with a text editor. In the print format file you define the entry fields to be printed and how they are arranged in the printout.

You can send your printout to an attached workstation printer or have the printout data sent to a file.

Before printing a dictionary or part of a dictionary, the output file structure must be determined. If you decide not to use one of the format files provided by **OpenTM2**, you must create this file in the subdirectory \eqf\prtform. For more information on how to use the sample format print files see [Defining the printout format](#). For a detailed description of the structure and syntax of a print format file see [Defining your own format file](#).

Prerequisites

- The dictionary must exist.
- The format file for the specific printout layout must exist.

Calling sequence

Select:

1. The "Dictionary List" window
2. The dictionary you want to print
3. **Print...** from the **File** menu

The "Print Dictionary" window (see [Figure 110](#)) is displayed.

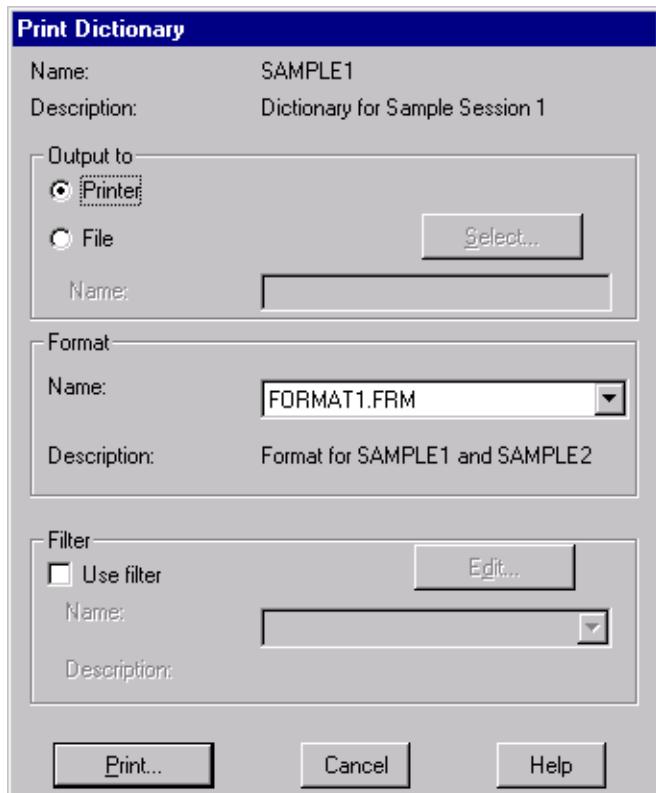


Figure 110. Print Dictionary window

Options and parameters

Name

The selected dictionary name.

Description

The dictionary description text.

Output to

In this box, select where you want to send your printout to:

Printer

If you want the printout on your attached workstation printer.

File

If you want the printout on file. Enter the complete file name specification—consisting of drive, path, file name—in the **Name** field yourself or click **Select...**, which takes you to the "Select File" window assisting you in selecting the drive, directory, and file name (see [Specifying an output file](#)).

Format

Specify the print format file to be used:

Name

Select a format file from the list of available files. If you created a new format file, it should appear in this list.

Description

Describes the selected format file.

Filter

Specify whether to use a filter. You can use an existing filter or change it or create a new filter.

Use filter

To use a filter, select this option and specify which filter to use:

Name

Lists all available filters. To use or change an existing filter, select it. To define a new filter, type the name in the list box.

Description

Describes the selected filter.

Click **Edit...** to define the new filter or to change an existing one, which takes you to the "Edit Dictionary Filter" window. For more information on how to define filter conditions, see [Defining a dictionary filter](#).

Click **Print** to start the printing function.

Results

If you selected the **Printer** option, the dictionary printout is sent to your attached printer. If you selected the **File** option, the dictionary printout is stored in the specified file.

Defining the printout format

OpenTM2 provides sample print-format files that you can use to print dictionary data or as a basis for defining your own print output format.

To become acquainted with format files, use the **FORMAT1.FRM** print format file for printing one of the sample dictionaries provided in a sample folder (for example **SAMPLE2**).

When you change a format file or create one of your own, make sure you follow the required syntax of the format file. All print-format files are located in the subdirectory **\eqf\prtform**.

The following sample files are provided by **OpenTM2**:

File name

Description

format1.frm

Prints each headword and its translation.

format2.frm

Prints all templates for each entry without formatting.

format3.frm

For this format file, it is recommended to select the **File** output option because it generates a BookMaster^(R) source file (containing all templates for each entry). This file must be processed by BookMaster^(R) outside **OpenTM2** to obtain a formatted dictionary printout.

format4.frm

Prints a dictionary extract consisting of cover page, body part, header section, and current page number for each page.

All the print-format files provided by **OpenTM2** are described in detail in [Sample format files](#). For a complete and detailed description of the structure and syntax of print-format files see [Defining your own format file](#).

Experienced users can use the print function also for generating new external dictionaries, which again can be imported and used in **OpenTM2**. For example, you can:

- Generate a reverse dictionary from an existing one (this means, generate a German to English dictionary from an English to German dictionary).
- Generate a new dictionary containing selected entries from another dictionary. This can be useful, for example, when you have a dictionary containing translations into several languages and want to extract the entry data related to translations into one of the languages.

Protecting a dictionary

dictionaryprotecting from uncontrolled modifications protecting dictionaries To avoid uncontrolled modification, you can protect dictionaries, that is, everyone can browse these dictionaries, but only authorized persons responsible for updating dictionaries (for example, a terminologist) can make changes to them.

dictionaryassigning a password You protect them by assigning a password to them. You can protect dictionaries in the "New Dictionary" window when creating a new dictionary or in the "Dictionary Properties" window when viewing its properties by pressing the **Protect...** button.

If you are not authorized to edit a protected dictionary but want to make changes to existing entries or add new entries, you can do this by creating a new dictionary using the protected dictionary as a model. For more details see [Creating a dictionary](#). You can then use the new dictionary as an addendum dictionary to the protected one. Redefine any dictionary search sequence so that this addendum dictionary precedes the protected dictionary.

You can send this addendum dictionary to a person who is authorized to edit protected dictionaries and can update these dictionaries on the basis of the modifications and additions made in your addendum dictionary. When you start creating a new dictionary, the unprotected mode is the default setting. In this mode you can carry out any modifications to that dictionary.

Dictionary task	Unprotected	Protected
Export to SGML-based dictionary	Yes	No
Delete a dictionary	Yes	Yes
Export folder with dictionary	Yes	Yes
Look up a dictionary entry	Yes	Yes
Edit a dictionary entry	Yes	No
Organize a dictionary	Yes	Password

The following table shows you which tasks you can perform with unprotected and protected dictionaries:

The following table shows the merge options that are available:

Dictionary merge from	To unprotected dictionary	To protected dictionary
SGML-based dictionary	Yes	Password
Unprotected folder dictionary	Yes	No
Protected folder dictionary	No	No

Calling sequence

To protect an existing dictionary, select:

1. The "Dictionary List" window
2. The dictionary you want to protect
3. **Properties...** from the **File** menu
4. **Protect** in the "Dictionary Properties" window

To protect a new dictionary, select:

1. The "Dictionary List" window
2. **New** from the **File** menu
3. **Protect** in the "New Dictionary" window

The "Protect Dictionary" window is displayed. Enter your new password twice and confirm by clicking **Protect**.

Results

The dictionary is protected, for example, only those that know the password have read-write access to the dictionary. Those who do not know the password have read-only access.

The protection status of a dictionary can be revoked if the password assigned to the respective dictionary is known.

Calling sequence

Select:

- The "Dictionary List" window
- The dictionary you want to unprotect
- **Properties...** from the **File** menu
- Click **Unprotect...**

The "Query Password for Dictionary" window is displayed.

This window is also displayed when you try to perform an action that makes changes to the dictionary.

In this window you can enter the dictionary password and confirm by pressing the **OK** button.

Results

The dictionary is unprotected and everyone can change it.

Searching for a dictionary entry

searching for a term in a dictionary dictionary If you want to know whether a certain term is included in a dictionary, you can search for this term in several ways, this means with a:

- Complete or incomplete term
- Predefined search type
- Dictionary filter defined by you
- Combination of these

Searching for a term can be done outside the **Translation Environment** and during translation in the **Translation Environment**.

To search for a term, you specify the term and the search parameters in the "Look up a Term" window.

If more than one term matches with the specified search criteria, you can select the term you are currently interested in to display its data.

If the term is found in the dictionary, you can display and edit the complete entry in subsequent windows (**Look up Entry** and **Edit Entry in Dictionary**).

If the specified term is not found in the dictionary, you can add a new entry for this term. What you can do in the following windows is described in [Looking up a dictionary entry](#), and [Editing a dictionary entry](#).

Prerequisites

The dictionary must exist.

Calling sequence

To search for a term outside the **Translation Environment**, select::

1. The "Dictionary List" window
2. The dictionary you want to open
3. **Open** from the **File** menu

Alternatively, double-click the dictionary in the "Dictionary List" window in the main window.

To search for a term in the **Translation Environment**, select:

1. A term or a multiword term in the "Translation" window by marking it
2. **Look up a term...** from the **Translate** menu.
If the term exists in the dictionary, the "Look up Entry" window for this term is displayed. For more information, see "**Looking up a dictionary entry**". If the term does not exist in the dictionary, **OpenTM2** starts searching for the term as if you specified the search option.

The "Look up a Term" window is displayed (see [Figure 111](#)).

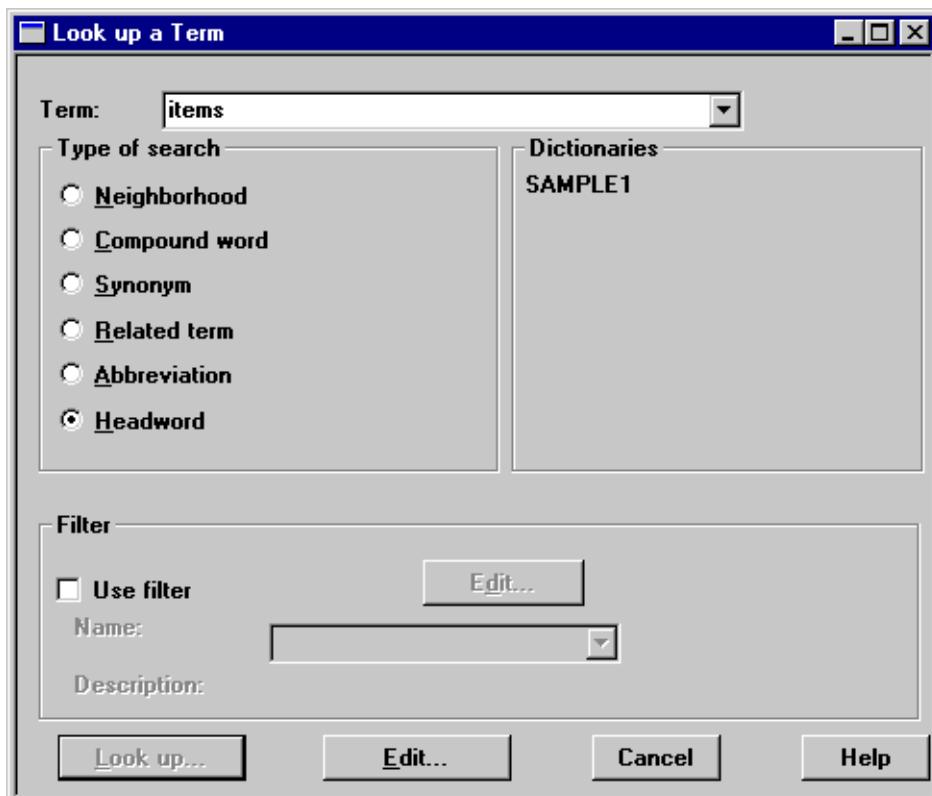


Figure 111. Look up a Term window

Options and parameters

Term

If you came from the "Dictionary List" window, type the term you are looking for. You can type a single term or a multiword term. You can use an asterisk (*) to get a list of all terms in the dictionary, or use the asterisk as part of a term to get a list of terms. For example, enter `app*` to get a list of all terms starting with the letters `app`. You can enter the term in uppercase, lowercase, or mixed-case characters.

If you came from the Translation Environment, the term that you highlighted or on which you positioned your cursor is displayed and a list of up to 30 terms preceding and following the term in alphabetical order.

Type of search

The synonym, related term, and abbreviation search types are greyed out if the corresponding entry field does not exist in the dictionary.

Select one of the following search types:

Neighborhood

To display the 30 terms preceding and the 30 terms following the specified term.

Compound word

To retrieve all compound words that start with the specified term. For example, if the term **data** is looked up, the possible compound words that would be retrieved are:
The displayed list is empty if no matches are found in the dictionary.

Synonym

To retrieve all terms where the specified term is in the synonym entry field of the respective term. For example, if you specified the term **stream** with the **Synonym** search option, you would get **brook** and **rivulet**, if **stream** is contained in their dictionary entries. The displayed list is empty if no matches are found in the dictionary.

Related Term

To display all terms related to the selected term. For example, if you specified the term **forest** with the **Related Term** search option, you would get **forestry commission** and **forester** as related terms if forest was entered in the related term entry field of their dictionary entries. The displayed list is empty if no matches are found in the dictionary.

Abbreviation

To display all terms that have an abbreviation. For example, if you specified the term **appl.** with the **Abbreviation** search option, you would get **application** if **appl.** was entered in the abbreviation entry field of its dictionary entry. The displayed list is empty if no matches are found in the dictionary.

Headword

To search only for the specified term. This is the default search option.

Dictionaries

This box displays the name of the selected dictionary if you came from the "Dictionary List" window, or the name of one or more dictionaries used during the translation of the document if you came from the Translation Environment.

Filter

Specify whether to use a filter to restrict the number of terms to be found. You can use an existing filter, change it, or create a new filter.

Use filter

To use a filter, select this option and specify which filter to use:

Name

Lists all available filters. If you want to use or change an existing filter, select it. If you want to define a new filter, enter a name in the list box.

Description

The description of the selected filter.

Click **Edit...** to define the new filter or to change an existing one, which takes you to the "Edit Dictionary Filter" window. For more details on how to define filters, see [Defining a dictionary filter](#).

Click **Look up...** to start the search.

Click **Edit...** to display the "Edit Entry in Dictionary" window in which you can edit the selected term, if it already exists in the dictionary, or define it to the dictionary as a new term.

Results

If you selected the **Headword** search type and an entry for the specified term is found in the dictionary, the "Lookup Entry" window is automatically displayed containing the data for this entry. For more information, see [Looking up a dictionary entry](#).

If you selected the **Headword** search type and the term is not found in the dictionary and you clicked **Look up...**, a message appears asking if you want to add the term to the dictionary. If you add the term, you are automatically taken to the "Edit Entry in Dictionary" window. For more information, see [Editing a dictionary entry](#).

If you selected **Neighborhood** search, and the term is not found in the dictionary, a list of up to 60 terms is displayed in alphabetical order where the specified term would appear in the middle. If more than one term satisfies the specified search criteria, a list of up to 60 terms is displayed from which you can select a term and use it for a **Headword** search request.

If the specified term exists in the selected dictionary and you selected **Edit...**, the "Edit Entry in Dictionary" windows is displayed. For more information, see [Editing a dictionary entry](#).

Viewing the properties of a dictionary

dictionarydisplaying properties dictionaryproperties The main properties of a dictionary are:

Name

The name of the dictionary. It cannot be changed for an existing dictionary.

Source language

The source of the dictionary, this means, the language of the terms (headwords). It cannot be changed for an existing dictionary.

Drive

The drive where the dictionary resides. It cannot be changed for an existing dictionary.

Description

The description of the dictionary. It can be up to 255 characters long.

Location

Can be **local** (only one translator at a time can use it) or **shared** (several translators can use it simultaneously).

Entry fields

Contains the entry fields that make up the dictionary structure. For an existing dictionary you can only change the name of a field and the lookup panel number.

Protected?

Specifies whether the dictionary is protected against unauthorized modifications.

Prerequisites

The dictionary must exist.

Calling sequence

Select:

1. A dictionary from the "Dictionary List" window
2. **Properties Summary...** from the **File** menu

Results

A document in HTML format is displayed containing a summary of the dictionary properties (see [Figure 112](#)).

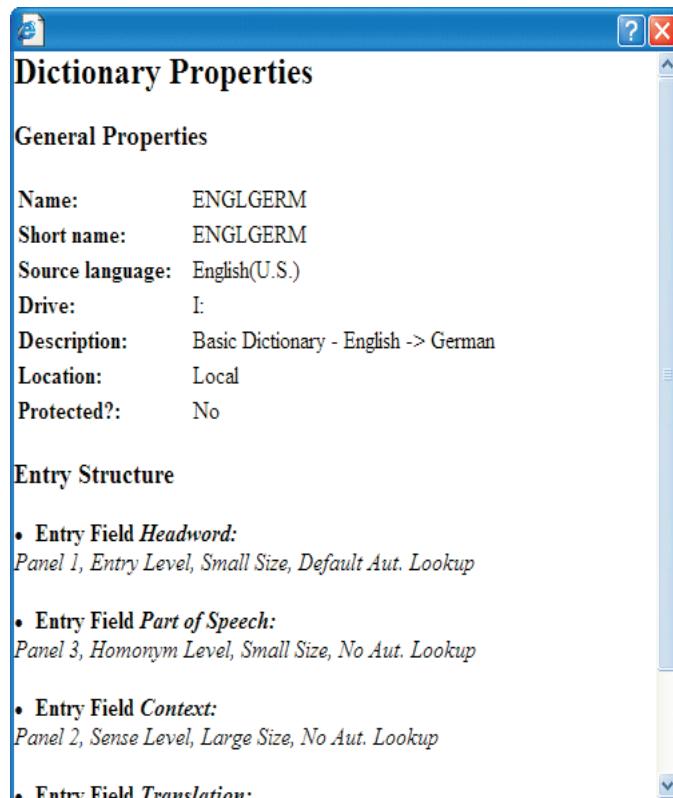


Figure 112. Dictionary Properties document

[PIC]Figure shows Dictionary Properties document

If you want to view more properties or change the properties of a dictionary, refer to [Changing the properties of a dictionary](#).

Changing the properties of a dictionary

The properties of a dictionary are its individual characteristics. Some properties, for example the dictionary structure and the size of an entry field, cannot be changed when the dictionary has been created. Others, for example which lookup panels are to display which data, can be changed.

Prerequisites

The dictionary must exist.

Calling sequence

Select:

1. The "Dictionary List" window

2. The dictionary from the list of dictionaries
3. **Properties...** from the **File** menu

This takes you to the "Dictionary Properties" window (see [Figure 113](#)).

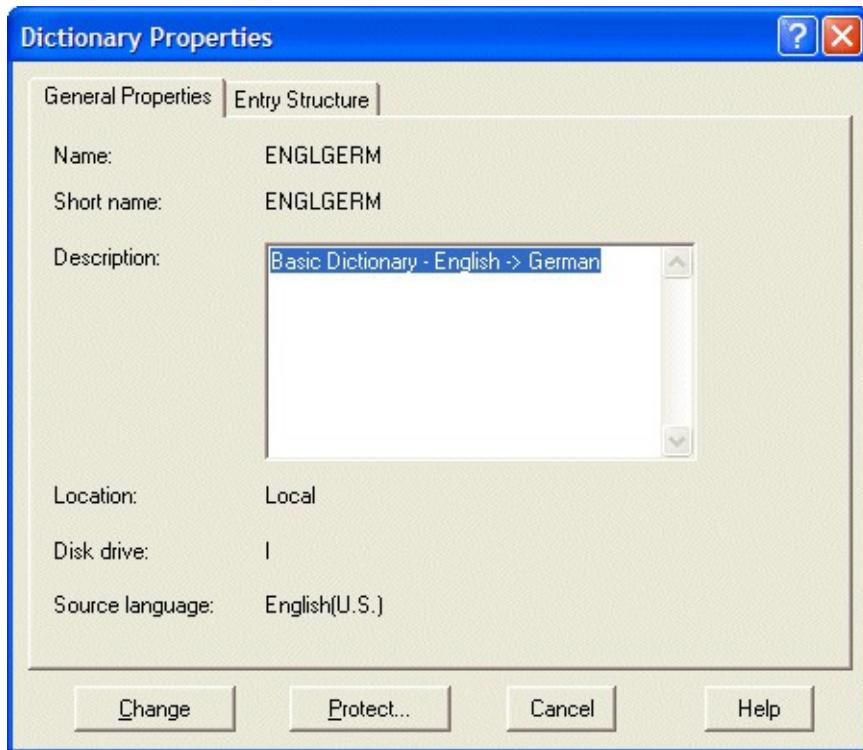


Figure 113. Dictionary Properties window

Options and parameters

The "Dictionary Properties" window lists the characteristics of the selected dictionary.

For an existing dictionary, only the following properties can be changed:

- The dictionary description
- The lookup display panel of an entry field
- The name of an entry field

On the "General Properties" page:

Name

The name of the dictionary. It cannot be changed for an existing dictionary.

Short name

The short name of the dictionary.

Description

The description of the dictionary. It can be up to 255 characters long.

Location

Can be **local** (only one translator at a time can use it) or **shared** (several translators can use it simultaneously).

Disk drive

The drive where the dictionary resides. It cannot be changed for an existing dictionary.

Source language

The source of the dictionary, this means, the language of the terms (headwords). It cannot be changed for an existing dictionary.

On the "Entry Structure" page:

Entry fields

Contains the entry fields that make up the dictionary structure. For an existing dictionary you can only change the name of a field and the lookup panel number.

Display

Click the **Panel** number on which you want this entry field displayed in the "Lookup Entry" window.

Omit means you do not want this entry field to be displayed at all. The entry field is not deleted from the entry structure. **Aut. Lookup** displays the complete entry field information in the window showing the found dictionary proposals during translation.

Level

Shows the level of the selected entry field. It cannot be changed for an existing entry.

Size

Shows the size of the selected entry field. It cannot be changed for an existing entry.

To change the name given to an entry field click **Change name....**. For example, you can change the name of the *Headword* entry field to *Term*. For more information on renaming entry fields see [Renaming a dictionary entry field](#).

To avoid unauthorized modification of a dictionary, you can protect it by clicking **Protect** and typing a password in the "Protect Dictionary" window. Then the dictionary can only be modified when the correct password is entered. For more detail on how to protect and unprotect dictionaries see [Protecting a dictionary](#).

Click **Change** to change dictionary properties.

Results

The dictionary properties are changed according to your specifications.

Viewing the details of a dictionary

In its simplest form, the "Dictionary List" window lists all of the available dictionaries. However, if you want to see more specific details of each dictionary, you can change your view of the "Dictionary List" window to have it display individual properties, such as the source language, location, or description of a dictionary.

The details selectable for a dictionary are:

Name

The name of the dictionary.

Description

The description of the dictionary.

Drive

The drive on which the dictionary resides.

Owner

Shows who created the dictionary (if it is a shared one). If it is a local dictionary, *n/a* is displayed.

Source language

The source language of the dictionary, this means the language of the headwords.

You can toggle between two display modes:

- Display the dictionary names only
- Display the names and details

You can change the arrangement of the dictionary details in the "Change Details" window.
For a general description see [Viewing and changing the details of a list item](#).

Option to select dictionary lookup for compound words in editor

User Interface

The new option can be activated or de-activated using the "Show single words of compound terms in auto-lookup" option of the Dictionary tab of the "Profile settings" dialog of the TranslationEnvironment:

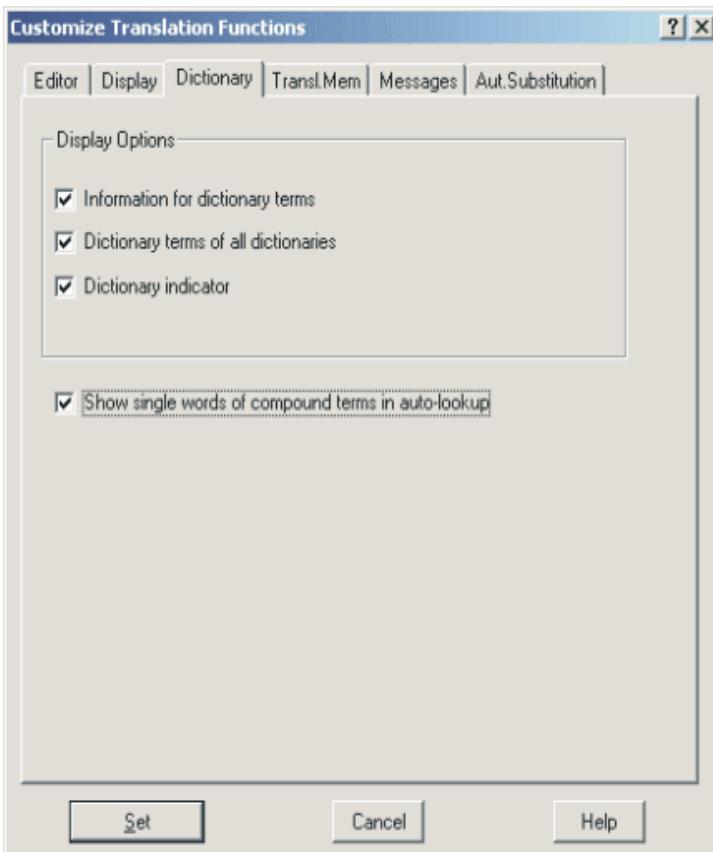


Figure 114. Dictionary Lookup
[\[PIC\]](#)Figure shows Dictionary Lookup

If the option is active the single words of a compound term are looked up together with the compound term. If the option is inactive the single words of a compound term are not looked up in the dictionary auto-lookup.

An example:

Assume a dictionary containing the terms "data", "data processing", "data processing unit", and "unit". If a segment contains the term "data processing unit" the dictionary window will show the translations for "data", "data processing", "data processing unit", and "unit" if the option is active and "data processing unit" if the option is inactive..

Viewing style indicators for dictionary terms

The style of dictionary terms can be specified using the "Style" field (<estyle>) on sense level and the "Trans Style" field (<tstyle>) on target level.

The following style values are predefined:

- Positive style values: "preferred"
- Negative style values: "deprecated", "non standard", and "no longer used"

When a term with predefined style information is displayed in the dictionary lookup window of the TranslationEnvironment, a style indicator is displayed before the term for styles on sense level and before the term translation for styles on target level.

For the positive style values this indicator is a plus sign ('+') with green background. The indicator for negative style values is the minus sign ('-') with red background.

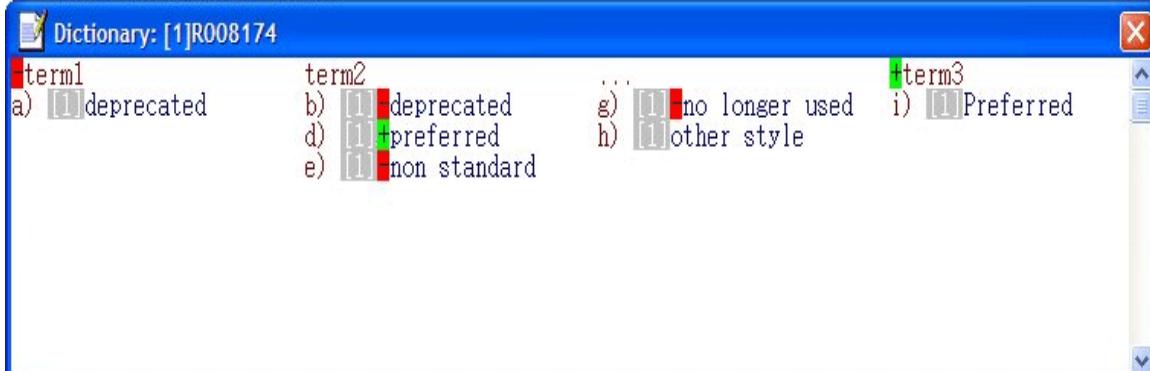


Figure 115. Style indicators for dictionary terms

Working with terminology lists

You use terminology lists to tell **OpenTM2** to look for or ignore specific terms when checking a document.

OpenTM2 creates terminology lists during document analysis. The lists are compiled by comparing terms in a document with selected dictionaries.

In **OpenTM2**, you can generate and work with the following types of terminology lists:

New terms lists

new terms list New terms are terms in a document that are not contained in specified dictionaries. You can use new terms lists to create a project-specific dictionary, to create your personal dictionary with terms that are often used for specific terminology, or to extend an existing dictionary.

Found terms lists

found terms list Found terms are terms in a document that are also contained in specified dictionaries. You can use found terms lists to copy existing entries from one dictionary to another or to check the used terminology in a dictionary before you use the dictionary for translation.

Exclusion lists

exclusion list Exclusion lists contain terms that **OpenTM2** can ignore when searching for new and found terms. Exclusion lists can be generated by copying selected new terms and found terms, or by editing in **OpenTM2** or outside the system. By using such an exclusion list, you can reduce the number of terms to be included the next time you create a new terms list or a found terms list. Exclusion lists usually contain so-called noise words. For each installed language support, **OpenTM2** already provides such a list.

A terminology list usually contains terms of only one language. When you copy a term from a terminology list to a dictionary, the source language of the dictionary must be the same.

Overview and terminology

All types of terminology lists can be *exported*, this means they are converted to an *external* format that is SGML based. A file with this format can be used to exchange the data of a list with other users or applications.

External terminology lists can be edited or created by other systems; to use them in **OpenTM2**, you *import* them.

A description of the external format is given in [Creating an external terminology list](#).

What you can do with terminology lists

You can start most of the functions related to terminology lists from the related list window, as shown in the following example.

Prerequisites

None.

Calling sequence

Select:

1. Utilities from the action bar
2. **Display terminology lists** from the menu
3. One of the list types offered in the cascaded window: **Exclusion Lists**, **Found Terms Lists** or **New Terms Lists**.

One of the list windows is displayed, for example the "New Terms Lists" window. The list window contains the list of existing terminology lists, in this example all new terms lists. Actions on a terminology list can be started by selecting a list name and an action from the **File** menu.

Creating a terminology list

Depending on the type of terminology list to be created, there are different ways to do this:

- New terms lists or found terms lists are usually created during the analysis of a document. An overview of how to do this is given in ["Analyzing documents using dictionaries"](#).
- Exclusion lists can be created in a **OpenTM2** window as described in ["Creating an exclusion list"](#).

You can also create an exclusion list from an existing new terms list or found terms list. This is

explained in [Opening and changing a terminology list](#).

Another general way is to create an external terminology list outside of **OpenTM2**, which can be imported. This is explained in [Creating an external terminology list](#).

Creating a list of new terms or found terms

terminology listscreating lists of new terms terminology listscreating lists of found terms
creatinglists of found terms creatinglists of new terms new terms listcreating found terms
listcreating New terms lists and found terms lists are created during analysis of a document.

Prerequisites

- At least one document must exist.
- At least one dictionary must exist.

Calling sequence

Select:

1. The document in the "Document List" window
2. **Analyze...** from the **File** menu
3. **Create lists** in the "Analyze Documents" window
4. **Create list of new terms** or **Create list of found terms** in the "List Settings" window
5. Any dictionaries to be used
6. **Set** to return to the previous window
7. **Analyze** to begin the analysis and the creation of the selected terminology lists

Options and parameters

For a detailed description of the options see [Analyzing a document](#).

Results

The terminology lists are created according to your specifications.

Creating an exclusion list

creatingexclusion lists terminology listscreating exclusion lists exclusion listscreating You can create an exclusion list from an existing new terms list or found terms list. This procedure is explained in [Opening and changing a terminology list](#).

You can also create an exclusion list from the "New Exclusion List" window or the "Edit Exclusion List" window.

Prerequisites

The "Exclusion Lists" window is active.

Calling sequence

Select **New...** from the **File** menu. The "New Exclusion List" window is displayed.

Options and parameters

The upper list box is empty. Here you can define the terms for the new exclusion list:

- To add a term, press Enter and type a term in the empty line.
- To delete a term already entered, mark it with the mouse and press Enter.
- To change a term already entered, overtype it. Pasting from the clipboard is also supported.

Name

Type the name of the new exclusion list.

To save the terms in the new exclusion list, select the **Create** button. If you do not want to create the exclusion list anymore, select the **Cancel** button.

Results

If you selected the **Create** button, the exclusion list is created with the terms entered and you can use it in the analysis of a document. If you selected the **Cancel** button, no new exclusion list is created.

Creating an external terminology list

Although **OpenTM2** offers functions for the creation of terminology lists where you only specify some parameters in a window, you can also create them outside of **OpenTM2**.

This can be useful when you already have terminology lists available in a format of your own, which you must only convert to the format you can import in **OpenTM2**.

External new terms lists and external found terms lists must be created in an SGML format to import them.

External exclusion lists are created without SGML tags. If you want to create an exclusion list outside of **OpenTM2**, you can write a list of terms in an editor of your choice, where each line contains one term, and then import it in external format. The list can have any number of terms. For details on importing terminology lists see [Importing a terminology list](#).

A new terms list requires as first tag in the first line `<NTLIST>` and as the very last tag in the last line of the list the tag `</NTLIST>`. The same applies to a found terms list, except that you must use the tags `<FTLIST>` and `</FTLIST>`.

The description of an entry always starts with the start tag `<TERM>` and ends with the corresponding end tag `</TERM>`.

The term itself is between the start tag `<LEMMA>` and the end tag `</LEMMA>`.

To add context information to a term, the context information must have the start tag `<CONTEXT>`, and the end tag `</CONTEXT>`. Adding context information is optional; you can include more than one context.

To insert the term **address** to a terminology list without context information, your entry would look as follows:

```
<TERM>
  <LEMMA>address</LEMMA>
</TERM>
```

When you want to insert the term **hardware** to a terminology list with information about two contexts, your entry would look as follows:

```
<TERM>
  <LEMMA>hardware</LEMMA>
  <CONTEXT>However, before selecting your hardware devices,
  you should understand your data processing requirements. </CONTEXT>
  <CONTEXT>This publication assists you in selecting
  a hardware configuration. </CONTEXT>
</TERM>
```

You can insert as much context information as you want, but each piece of context information needs the start and the end tag.

A new terms list with the entries **hardware**, **software**, **term**, **context**, and with some context information for the first and last entry would have the following layout:

```
<NTLIST>
  <TERM>
    <LEMMA>hardware</LEMMA>
    <CONTEXT>Your hardware requirements are as follows:</CONTEXT>
  </TERM>
  <TERM>
    <LEMMA>software</LEMMA>
  </TERM>
  <TERM>
    <LEMMA>term</LEMMA>
  </TERM>
  <TERM>
    <LEMMA>context</LEMMA>
    <CONTEXT>The context information must be stored between a start
    and end tag.</CONTEXT>
  </TERM>
</NTLIST>
```

You do not have to sort your entries in alphabetical order. This is done during importing. How to import the new list is described in [Importing a terminology list](#).

Deleting a terminology list

Prerequisites

terminology lists removing deleting terminology lists The list window that contains the terminology list to be deleted is active.

Calling sequence

Select:

1. The terminology list to be deleted
2. **Delete** from the **File** menu

Options and parameters

Before the system deletes the selected list, it asks you to confirm whether you really want it:

- If you select **No**, you can leave the delete function.
- If you select **Yes**, the list is deleted.

Results

If you selected **Yes**, the list is deleted; otherwise it remains unchanged.

Exporting a terminology list

exportingterminology lists terminology listsexporting You can export a terminology list to give it to another user, for example, another translator who needs it for the translation of another document. You can choose to export the list in **OpenTM2** or external format.

In order to exchange the data with other **OpenTM2** users, select the **OpenTM2** format. If you want to pass the list to another program, you should select the external format.

When you export in external format, the found terms lists and the new terms lists are exported in SGML format. You can find a description of this format in [Creating an external terminology list](#).

When you export an exclusion list in external format, it is written as an ASCII-coded file.

Prerequisites

The list window that contains the terminology list to be exported is active.

Calling sequence

Select:

1. The terminology list to be exported
2. **Export...** from the **File** menu

For an exclusion list, the "Export Exclusion List" window is displayed (see [Figure 116](#)).



Figure 116. Export Exclusion List window
[\[PIC\]](#)Figure shows Export Exclusion List window

Options and parameters

Depending on the format selected, this window offers a different number of options and parameters. For exporting in **OpenTM2** format, specify the following:

Format

Select **OpenTM2**.

To Select the drive to which you want to export the terminology list.

For exporting in external format, specify the following:

Format

Select **External**.

To Specify the needed parameters where to store the terminology list and select the drive to which you want to export the list.

Name

Define the file name for the exported list.

Current directory

Displays the directory selected from the **Directories** list box.

Directories

Contains all available directories on the drive selected. Select the directory where to store the exported list.

To start the exporting function, select **Export**.

Results

The selected terminology list is now exported in the selected format. The file in the internal format can only be used by **OpenTM2** users. The file in the external format can be used or changed by other applications. In [Creating an external terminology list](#) you find a description of how to work with external lists.

Importing a terminology list

importing terminology lists terminology listsimporting If a terminology list has been created by another **OpenTM2** user, you must import it to make it available in **OpenTM2**.

Prerequisites

To import a new terms list, the "New Terms Lists" window must be active.

- To import a found terms list, the "Found Terms Lists" window must be active.
- To import an exclusion list, the "Exclusion Lists" window must be active.

Calling sequence

Select **Import...** from the **File** menu.

For an exclusion list, the "Import Exclusion List" window is displayed (see [Figure 117](#)).

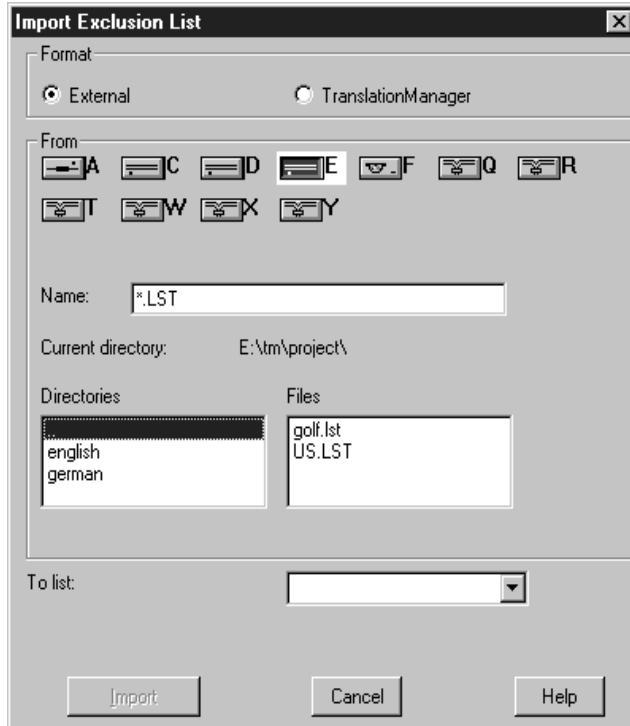


Figure 117. Import Exclusion List window
[PIC]Figure shows Import Exclusion List window

Options and parameters

Depending on the format selected, this window offers a different number of options and parameters:

For importing in **OpenTM2** format, specify the following:

Format

Select **OpenTM2**.

From

Select the drive from which you want to import the terminology list. When you have selected a drive the terminology lists are automatically displayed in the list box. Select the list you want to import.

To list

Type the name under which you want the data stored in **OpenTM2**.

For importing in external format, specify the following:

Format

Select **External**.

From

OpenTM2 needs information where the terminology list to be imported currently resides: Select the drive where the external list is stored.

Name

Type the name of the external list.

Current directory

Displays the directory selected from the **Directories** list box.

Directories

Select the directory containing the external terminology list from the list of directories on the selected drive.

Files

Select the file name of the external terminology list from the list of names in the currently selected path.

To list

Type the name under which you want the data stored in **OpenTM2**.

To start the importing function, select the **Import** button.

Results

The terminology list is stored in **OpenTM2** and can be used for further processing.

Opening and changing a terminology list

terminology lists modifying terminology lists moving terms to another list opening terminology lists You open a list to work with it, for example, to copy entries from the terminology list to a dictionary or to another list.

Prerequisites

The list window that contains the terminology list to be opened is active.

Calling sequence

Select:

1. The terminology list to be opened
2. **Open** from the **File** menu

One of the following windows is displayed, depending on the type of list selected:

- "Work with Found Terms List"
- "Work with New Terms List"
- "Edit Exclusion List"

See [Figure 118](#), the "Work with New Terms List" window, as an example.

The "Edit Exclusion List" window is explained in [Editing an exclusion list](#).

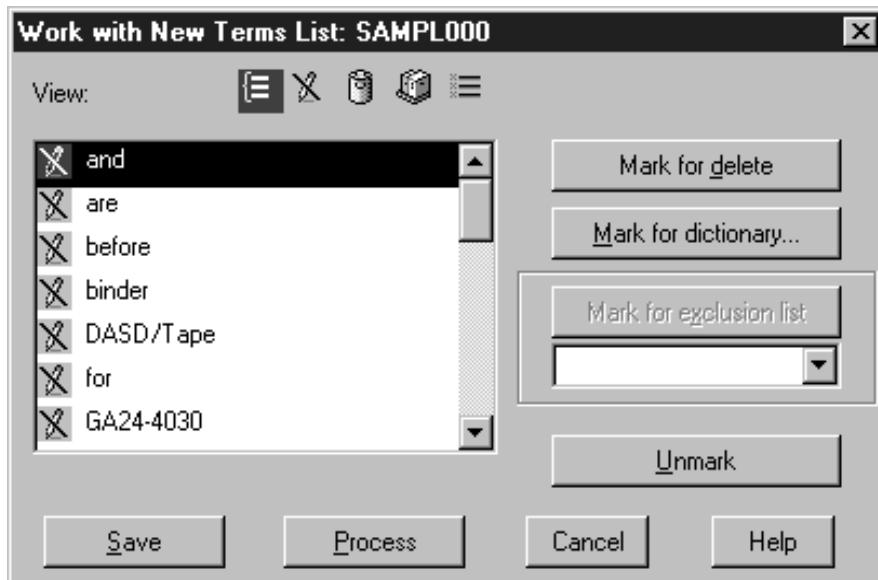


Figure 118. Work with New Terms List window
[\[PIC\]Figure shows Work with New Terms List window](#)

Options and parameters

In the list box you see the terms contained in the opened new or found terms list.

View

The meaning of the processing symbols in this row is as follows:

- [\[PIC\]Figure shows](#)

Shows all terms in the list.

- [\[PIC\]Figure shows](#)

Shows all terms not marked for any processing.

- [\[PIC\]Figure shows](#)

Shows all terms to be deleted.

- [\[PIC\]Figure shows](#)

Shows all terms to be added to a dictionary.

- [\[PIC\]Figure shows](#)

Shows all terms to be added to an exclusion list.

The contents of the list box depends on the processing selected for the terms and the processing symbol selected in this row.

You can select and mark one or more terms in order to specify the processing:

Mark for delete

Select this button to remove this term from the lists.

Mark for dictionary...

Select this button to add this term to a dictionary. This takes you to the "Mark Term for Dictionary" window where you specify further parameters and then return to this window (see [Marking a term for a dictionary](#)).

Mark for exclusion list

Select this button to add this term to an exclusion list. You can select the exclusion list to be filled from the list box, which contains all available exclusion lists. You can also type a new name for an exclusion list in order to create a new one.

Unmark

Click this option to unmark a previously marked term.

The view symbol preceding a term represents the processing you have selected for it.

To begin the processing of the selected terms, you click **Process**. If you marked any term to add to a dictionary, you can specify more options in the "Mark Term for Dictionary" window (see [Editing a dictionary entry](#)). When the processing is completed, the list contains the remaining terms for which no processing was specified.

The marked terms need not be processed immediately. To save the list in its current status of marking, click **Save**.

To leave the terms list in its original status, click **Cancel**.

Results

If you selected **Process**, the terminology list is changed depending on your selections. If you made changes to any dictionary or exclusion list, they are active during the next analysis or translation of a document. If you selected **Save**, the terminology list is saved including the new processing marks. If you selected the **Cancel** button, the terminology list remains unchanged.

Marking a term for a dictionary

terminology lists moving terms to dictionary terminology lists marking terms for adding to dictionary
You can mark new terms and found terms to add them to a dictionary. You do this by clicking **Mark for dictionary...** in one of the following windows:

- "Work with New Terms List"
- "Work with Found Terms List"

The "Mark Term for Dictionary" window is displayed (see [Figure 119](#)).

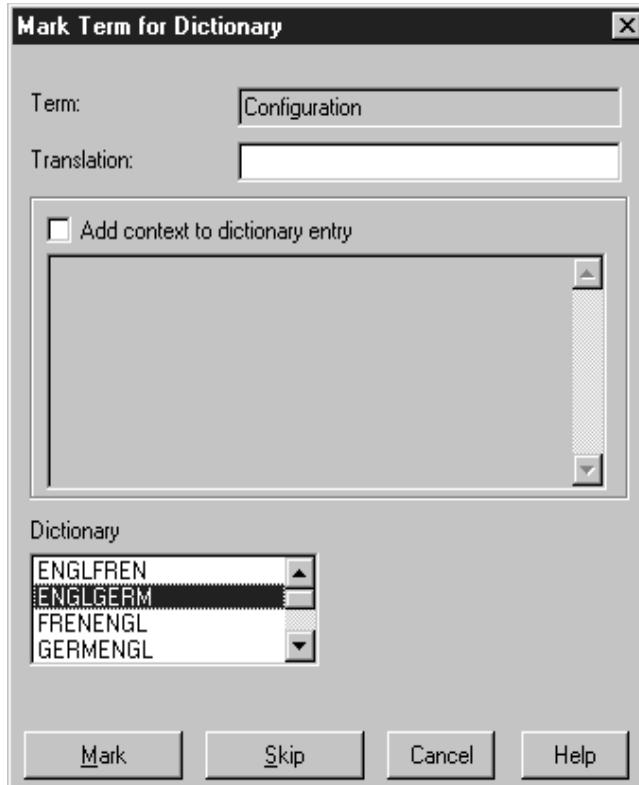


Figure 119. Mark Term for Dictionary window
[PIC]Figure shows Mark Term for Dictionary window

Options and parameters

Term

The current term from the terminology list is displayed.

Translation

Optionally, type the translation for the term to be copied to the dictionary.

Add context information to dictionary entry

If you have created the terminology list with context information, this information is copied to this field. You can also type your own context information. To save the context information to the dictionary, mark the check box.

Dictionary

The list box contains all available dictionaries that are not protected. Select the dictionary where you want the term to be added.

To save the processing information for this term, click **Mark**.

To skip the current term, select the **Skip** button.

If several terms are marked for the dictionary, the next term is displayed in the **Term** field. If all marked terms were displayed, you are returned to the previous window.

Results

The selected term is prepared for addition to a dictionary. When all terms marked for the dictionary are processed, you can add more information, depending on the structure of the selected dictionary in the "Edit Entry in Dictionary" window. You can find more information about editing entries in dictionaries in [Editing a dictionary entry](#).

Editing an exclusion list

terminology listsmodifying an exclusion list exclusion listsadding a term exclusion listschanging a term exclusion listsremoving a term editingexclusion lists When you have opened an exclusion list, the "Edit Exclusion List" window is displayed.

Options and parameters

The list box contains terms from the selected exclusion list.

- To change a term, overtype it.
- To add a term, press Enter and type a term in the empty line.
- To delete a term, mark it and press Enter.

Clipboard operations (cut, copy, paste) are supported.

Save to

Displays all the exclusion lists to which terms can be saved. Select an existing exclusion list or overtype it with another name (to create a new list).

To save the terms in the selected exclusion list, select the **Save** button.

To leave the exclusion list in its original status, select **Cancel**.

Results

If you selected **Save**, the exclusion list is changed depending on your selections. If you made changes to the exclusion list, they are active during the next analysis of a document.

If you selected **Cancel**, the exclusion list remains unchanged.

Printing a terminology list

terminology listsprinting printingterminology lists To do paperwork on terminology lists, you can print them on the printer attached to your workstation.

Prerequisites

The respective list window that contains the terminology list to be printed is active.

Calling sequence

Select:

1. The terminology list to be printed
2. Print from the **File** menu

The printing function is started immediately. The respective list window remains displayed.

Options and parameters

None.

Results

The selected terminology list is directly routed to your attached printer. The first line of the printout contains the type of terminology list and the list name. Then the contents of the list follow. Each entry is printed in a separate line. If your list contains context information, it is also printed.

Viewing the details of a terminology list

For all types of terminology lists, the system keeps some view details that can be displayed in the corresponding list window.

These details are:

Name

The name of the list.

Drive

The drive the list resides on.

Last Update (Date)

The date when the list was last changed.

Last Update (Date + Time)

The date and time when the list was last changed.

Size

The number of bytes that the list occupies on your disk.

You can toggle between two display modes:

- Display the names only
- Display selected details

You can change the arrangement of the details of terminology lists in the "Change Details" window described in [Viewing and changing the details of a list item](#).

Working with markup tables

OpenTM2 can handle documents that are created by the most popular word-processing systems and a wide range of other document types.

Documents usually are a mixture of text and layout information, which describes the formatting of the document. This layout information is called markup data. It defines, for example, the start of a new page, the start of a paragraph, or character layout like boldness, italics, or underline.

In most of the word-processing systems that display a text in WYSIWYG mode (What You See Is What You Get), you usually do not see this markup but only its effects. Some systems allow you to display and check the markup data, for example in WordPerfect with the **Reveal codes** option. In other systems that generate the formatting printout in a separate step, you usually edit the markup yourself.

The table that stores the description of the markup data is called *markup table* in **OpenTM2**. **OpenTM2** needs this markup table to differentiate between the text to be translated and the markup data.

Markup tables are used:

- During analysis to divide the document into translatable and nontranslatable parts
- During translation to protect the markup information, so that translators are warned if they try to overtype it

It is important for you to know which markup table is to be used for which type of document. You must tell **OpenTM2** what kind of markup table should be associated with a folder or a document. For more information on associating a markup table with a document see either [Changing the properties of a document](#), or [Importing a document](#).

For information on associating a folder with a markup table, see [Creating a folder](#), or [Changing the properties of a folder](#).

The association of a document or folder with a markup table is essential because the different word processors have different layout descriptions. All these layout descriptions are understood by **OpenTM2** and processed in a general way.

The following table shows the available markup table names and the supported document types.

markup tables delivered with OpenTM2	Type of document
Markup table	
EQFALINE	ANSI files in which everything is translatable.
EQFAMI	Ami Pro texts (Ami Pro for Windows ^(R) , Version 2.0).
EQFAMRI	ANSI files in which translatable text is enclosed in double quotes.
EQFANSI	ANSI documents. This markup table is combined with a user exit.
EQFAQOUT	ANSI files in which translatable text is enclosed in single quotes.
EQFASCII	ASCII files in which everything is translatable.
EQFASM	Assembler documents.
EQFBOOK	BookMaster ^(R) and Information Presentation Facility (IPF) texts.

markup tables delivered with OpenTM2	Type of document
Markup table	
EQFFRBLD	FrameBuilder files (Version 5.5).
EQFHTML2	HTML texts (level 2.0).
EQFHTML4	HTML texts (level 4.0). This markup table is combined with a user exit.
EQFINT2	Interleaf files.
EQFLINE	ASCII files in which everything is translatable.
EQFMRI	ASCII files in which translatable text is enclosed in double quotes.
EQFMSWRD	Microsoft (R) Word for Windows (R) documents.
EQFPPT	Microsoft (R) PowerPoint (R) documents.
EQFQUOTE	ASCII files in which translatable text is enclosed in single quotes.
EQFRF	Rich Text Format (RTF) files coming from Word for Windows (R), Ami Pro, or other word processors.
EQFSGML	SGML documents that were designed for processing by Lotus (R) Notes (TM). Also applies to help texts for programs running under Windows (R) that can be saved as RTF documents.
EQFWP	WordPerfect texts (Version 5.0).
EQFWP6	WordPerfect texts (Version 6.0 and 6.1).
EQFUTF8	Unicode document in UTF-8 encoding
LOTUSAMI	Ami Pro documents designed for processing by Lotus (R) Notes (TM).
LOTUSNGD	Lotus (R) Notes (TM) documents.
UNICODE	Unicode documents in UTF-16 encoding.

OpenTM2 allows you to add more formats. For detailed information on creating external markup tables see [Creating a markup table](#), and [Working with external markup tables](#).

For information on how to use the markup tables and how to prepare the different document types for translation, see [Dealing with specific document formats](#).

Overview and terminology

Markup tables can be exchanged with other users or applications. This may be necessary, when you have changed a markup table for your special translation needs and another translator may want to share your changes for similar translation environments.

To give the markup table to someone else you must transfer it out of **OpenTM2**. This process is called *export*. The recipient must *import* the exported markup table into **OpenTM2** again.

Markup tables can also be exchanged via exported folders. When you export a folder with a document using a specific markup table, this markup table is included in the exported folder and can be loaded implicitly during folder import in the recipient's system.

Markup tables can be changed. You can choose from two methods:
markup table
exporting **importing** **changing** **properties**

- One way is to *export* it from OpenTM2 and to use an external editor to apply the changes. You need to *import* the markup table into OpenTM2 to use it again, and you need to be familiar with the SGML-based format of the exported markup table.
- Another way is to *change the properties* of a markup table without leaving OpenTM2.

Each method has its advantages. See [Changing a markup table](#) for the details.

You can also create new markup tables, as described in [Creating a markup table](#). New markup tables also need to be imported into OpenTM2 before they can be used for translations.

What you can do with markup tables

Start the functions related to markup tables from the "Markup Table List" window.

Prerequisites

None.

Calling sequence

Select **Display markup table list** from the **Utilities** menu. The "Markup Table List" window is displayed. The window contains all existing markup tables in your system. Most of the actions related to markup tables can be started by selecting a list name and an action from the **File** menu, such as **Delete**, **Export...**, or **Import....**

Changing a markup table

A markup table can be changed by using two different methods:

- You can export an existing markup table from OpenTM2's internal format to an SGML-based external format, which can be edited with any text editor. The contents of an exported markup table are described in terms of SGML statements, so you need to be familiar with SGML. After changes are made you need to import the markup table into OpenTM2 before you can use it again. Follow these steps if you choose this method:
 - Export an existing markup table from OpenTM2, as described in "[Exporting a markup table](#)". This ensures that you start with a markup table that has the correct SGML syntax.
 - Make a backup copy. The original markup tables provided by OpenTM2 cannot be reinstalled.
 - Change and replace the markup data in the exported file as required, as described in "[Creating new markup tables](#)"
 - Before importing the external markup table, consider to rename the file to protect the original markup table until you are finished.
 - Import the external markup table, as described in "[Importing a markup table](#)" OpenTM2 performs a syntax check during the import. It does not import an incorrect markup table, so you need to correct it in its external format.
Note that an external markup table can be imported either under the old name or a new name.

The disadvantages of this method are that you need to export and import the markup table and that you need to know SGML. Further, no syntax checking is provided (other than OpenTM2 refusing to import an improperly coded markup table).

The advantage is that extensive changes are faster. You can also use tools of choice to automate changes to markup tables, for example word processing macros.

For more detailed information see [Working with external markup tables](#).

- Within OpenTM2, you can use the "Markup Table Properties" window to change, add, and remove properties of a markup table. The properties comprise all aspects of the content of a markup table. In addition, you can test your modifications immediately, and you can password-protect a markup table.

The advantages of this method are that you do not need to leave OpenTM2 to change a markup table and that less knowledge about SGML is required.

On the other hand, the graphical user interface might not be appropriate for extensive changes.

To actually change a markup table by means of the "Markup Table Properties" window, see [Changing the properties of a markup table](#).

Before you change a markup table provided by **OpenTM2**, it is recommended to make a backup copy of the original version because the original markup tables cannot be reinstalled.

Creating a markup table

New markup tables can be created by using two different methods:

- You can create an external markup table in SGML-based format and import it into OpenTM2.
- Within OpenTM2, you can use the "Markup Table Properties" window to create a new markup table.

Creating a new markup table is similar to changing an existing markup table. Therefore, see [Changing a markup table](#) for the details and advantages of both methods. Note that it is often more practical to modify a copy of an existing markup table.

Deleting a markup table

[deleting markup tables](#) If you no longer need a markup table, you can delete it. For markup tables provided by **OpenTM2**, it is recommended to create a backup copy by exporting the table, because it cannot be reinstalled separately but only by a system reinstallation.

Prerequisites

- The "Markup Table List" window must be active.
- The markup table to be deleted must be selected.

Calling sequence

Select **Delete** from the **File** menu.

Options and parameters

Before the system deletes the selected markup table, it asks you to confirm whether you really want to delete it.

If you select **No**, you will leave the delete function.

If you select **Yes**, the markup table will be deleted.

Results

If you selected **Yes**, the markup table is deleted. Otherwise it remains unchanged. References to this markup table may still exist in certain folders. Update the folder properties so that the deleted markup table is no longer associated with a folder.

Exporting a markup table

exportingmarkup tables markup tablesexporting You can export a markup table to give it to other users, for example, other translators who must use the same modified markup table in their translation environment.

Markup tables can only be exported in external format.

Each markup tag described in the markup table is exported together with its nondefault data. You can find a table that contains the definition of markup table SGML tags and the data defaults that are used for **OpenTM2** in [Working with external markup tables](#).

Prerequisites

- The "Markup Table List" window must be active.
- The markup table to be exported must be selected.

Calling sequence

Select **Export...** from the **File** menu.

The "Export Markup Table" window is displayed.

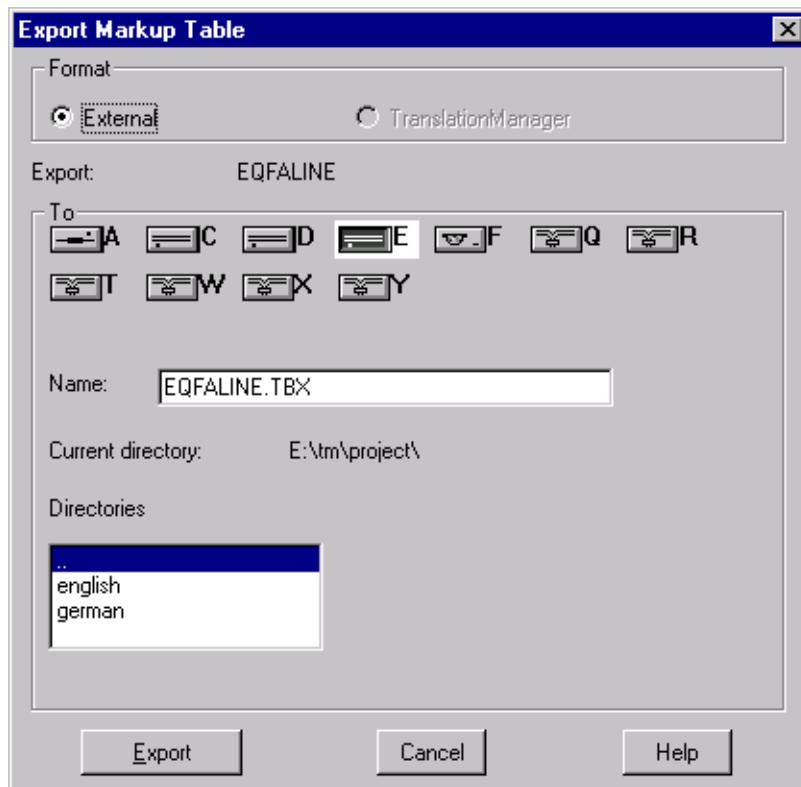


Figure 120. Export Markup Table window

Options and parameters

Format

External is selected automatically.

To

Enter the required parameters where to store the exported markup table:Select the drive to which you want to export the markup table.

Name

Define the file name for the exported markup table.

Current directory

Displays the directory selected from the **Directories** list box.

Directories

This list box contains all available directories on the selected drive. Select the directory where to store the markup table.

To start the exporting function, select the **Export** button.

Results

The selected markup table is exported to the specified file and saved on the selected drive and directory.

Importing a markup table

markup tablesimporting markup tablesmaking them available importingmarkup tables If another user has prepared a markup table for your translation environment, you must import it in order to be able to use it in **OpenTM2**.

Markup tables can only be imported in external format.

Prerequisites

The "Markup Table List" window must be active.

Calling sequence

Select **Import...** from the **File** menu. The "Import Markup Table" window is displayed (see [Figure 121](#)).

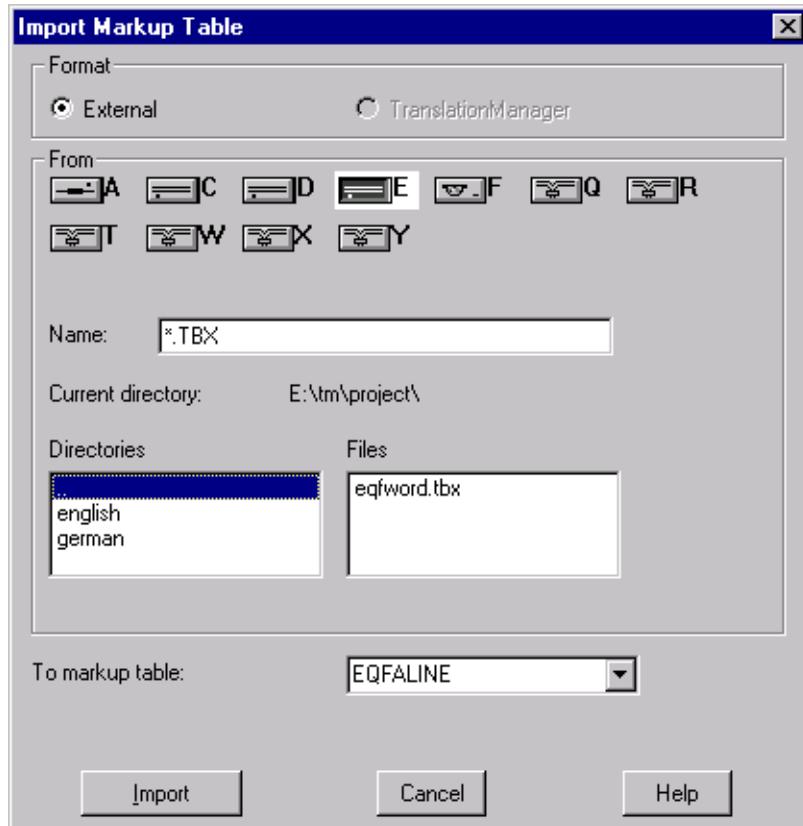


Figure 121. Import Markup Table window

Options and parameters

Format:

External is selected automatically.

From

Specify where the markup table to be imported currently resides and select the drive where the external markup table is stored.

Name

Type the name of the external markup table or select the name from the following list boxes.

Current directory

Displays the directory selected from the **Directories** list box.

Directories

Select the directory containing the external markup table from the list of directories on the selected drive.

Files

Select the file name of the external markup table from the list of names in the currently selected path.

To markup table

Contains the name of the selected markup table to be imported. You can overwrite this name to create a new markup table or select a name from the list box to merge your file into an existing markup table.

To start the importing function, select the **Import** button.

Results

The markup table is now available in OpenTM2, you can start to associate it with documents or folders.

Changing the properties of a markup table

markup tables changing properties of markup tables properties of markup tables The "Markup Table Properties" window allows you to change, create, test, and protect a markup table without the need to export the markup table to an external format. You can change an existing markup table, which includes the addition, removal, and modification of markup tag definitions and markup attribute definitions. You can create a new markup table from scratch (however, it might be more efficient to choose the external method, as described in [Creating a markup table](#) and [Changing a markup table](#)). You can dynamically test a markup table against a test document and continue to change it without leaving the "Markup Table Properties" window. And you can password-protect a markup table against unauthorized modifications.

Before you apply any changes to a markup table or attempt to create a new one, you should know its syntax and semantics, as described in [Creating new markup tables](#). Note that most options and parameters on the "Markup Table Properties" window pages have their equivalent SGML tags described in the subchapters of [Creating new markup tables](#).

Before you change a markup table provided by OpenTM2, it is recommended to create a backup copy by exporting the markup table.

Note:

The contents of markup tables are sensitive. Improper modifications avoid the correct handling of documents that are associated with them. Apply the password protection to prevent unauthorized modifications.

Prerequisites

The "Markup Table List" window must be active.

Calling sequence

To change an existing markup table:

- Select a markup table in the "Markup Table List" window.
- Select **New...** from the **File** menu.

To create a new markup table:

- Select **New...** from the **File** menu.

If an existing name of a markup table is chosen, a warning message is shown.

The "Markup Table Properties" window is displayed (see [Figure 122](#))

. windowsMarkup Table Properties window Markup Table Properties window

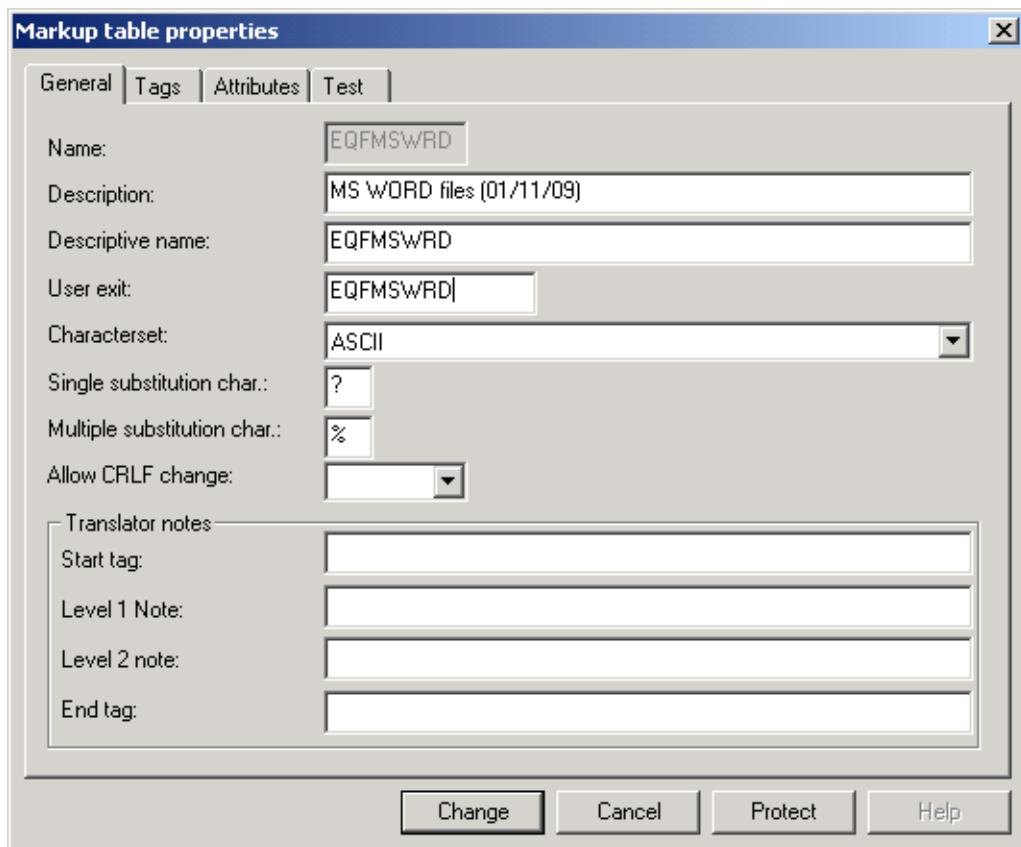


Figure 122. Markup Table Properties window

Options and parameters

On the "General" page you can set or change general markup table properties: [Markup Table Properties window](#)[General page](#)

Name

Contains the name of the markup table being changed. This field cannot be changed if an existing markup table is changed. The field accepts a markup table name if you started to create a new markup table.

Description

[markup table](#)[description](#) Contains a markup table description, which is shown in the "Markup Table List" window.

Descriptive name

[markup table](#)[descriptive name](#) Contains a descriptive name for this markup table. For example, if **Name** is **EQFBOOK**, you could simply describe it as **BOOK**. If you create a new markup table, the descriptive name must be unique.

User exit

[markup table](#)[user exit](#) Contains the name of the user exit, if the markup table uses one. If you create a new markup table, and you enter the name of a user exit that does not exist, a warning message is shown. You can confirm this message to continue.

Character set

[markup table](#)[character set used with](#) Contains the specified character set that is to be used for import and export of documents that use this markup table. You can specify **ASCII**, **ANSI**, **UNICODE**, or **UTF8**.

Single substitution character

[markup table](#)[substitution character used in](#) Contains the substitution character to use for single character substitution. The default character is **?**. If you create a new markup table, the character must be different from that used as multiple substitution character.

Multiple substitution character

markup tablessubstitution character used in Contains the substitution character to use for multiple character substitution. The default character is %. If you create a new markup table, the character must be different from that used as single substitution character.

Allow CRLF change

markup tablessubstitution character used in If set to NO, do not allow reflow in the editor.

Translator notes

markup tablestranslator's notes in The following fields contain the markup for translator's notes. Before changing or adding information here you should be familiar with the internals of a markup table and how translator's notes are added to a source document. For the latter see [Including notes for the translator](#). Note especially that the level 1 and level 2 translator's note tags are nested between the start tag and end tag. The latter two are usually the annotation or comment tags and have their own definitions in a markup table. If the translator's notes start and end tags do not match the annotation start and end tags, OpenTM2 fails to properly recognize translator's notes.

Start tag

markup tablestranslator's note start tag Contains the start tag that begins an annotation or comment in the source document. For example, :annot for a BookMaster document, or <!-- for an HTML document (but not :annot., respectively <!----).

Level 1 Note

markup tablestranslator's note level 1 note translator's note level 1 Contains the level 1 translator's note tag, for example, trnote1:

Level 2 Note

markup tablestranslator's note level 2 note translator's note level 2 Contains the level 2 translator's note tag, for example, trnote2:

End tag

markup tablestranslator's note end tag Contains the end tag that terminates an annotation or comment in the source document. For example, :eannot for a BookMaster document, or -- for an HTML document (but not :eannot., respectively -->).

On the "Tags" page you can work with markup tag definitions. On the left of this page you see the list of tags that are defined in this markup table. You can select a tag by clicking it. You can add new tags (by clicking **Add new tag**) and delete selected tags (by clicking **Delete tag**). On the right of this page you see the definition, respectively properties, of a selected tag. You can change the properties of a tag by changing the entry fields, by selecting or clearing the check boxes, and by selecting options from drop-down lists.

If you create a new markup tag, you must at least specify the tag string and its delimiter, respectively its length. [Markup Table Properties window](#)[Tags page](#)

Tag string

markup tablestag definitions in Contains the string that represents the markup tag.

End delimiter

Contains one or more characters that end the tag string. Two or more characters cause OpenTM2 to check for all possible combinations of these characters to determine the end of the tag. You cannot specify a string as end delimiter.

Alternatively, you can specify a length that specifies the end of the tag string (see parameter **Length**).

Length

Contains the length of the tag string. Alternatively, you can specify end delimiters that specify the end of the tag string (see parameter **End delimiter**).

Column position

Contains the column position where the markup tag starts. If a markup tag has no special start position and can occur anywhere in a line, this parameter is omitted or can be set to 0. The default is 0.

Tag type

Contains the type of the markup tag, which specifies whether the tag starts or ends a segment and whether the tag influences OpenTM2's segmentation. See "TYPE" in [Table 1](#) for a description of possible selections in the drop-down list.

Segmentation flag

Contains a flag for this markup tag that controls OpenTM2's segmentation and text protection.

- If segmentation is toggled off, text following this markup tag is not segmented by OpenTM2 until it encounters a markup tag that toggles on segmentation.
- If text protection is toggled on, text following this markup tag is protected by OpenTM2 until it encounters a markup tag that toggles off text protection.

See "SEGINFO" in [Table 1](#) for a description of possible selections in the drop-down list.

Text processing flag

Contains a flag for this markup text that specifies how text following this tag is associated with the tag. See "ASSTEXT" in [Table 1](#) for a description of possible selections in the drop-down list.

Unique ID

Contains a unique ID in the range from 1 to 65565 that is associated with this tag. The use of these identifiers is at your disposal.

Additional information

Specifies whether specific text is to be ignored when segments are aligned during the creation of an Initial Translation Memory. See "ADDINFO" in [Table 1](#) for a description of possible specifications.

Class ID

Contains a class identifier. **CLS_HEAD** causes this markup tag to become an entry of the table of contents that you can display during the translation of a document using the **Special go to...** dialog.

Tag has attributes

This check box is selected if this markup tag has attributes attached (which requires that these attributes are listed on the "Attributes" page). This parameter is equivalent to the specification of "ATTRINFO" in [Table 1](#).

Can contain translator's notes

This check box is selected if this markup tag can contain translator's notes. See [Translator's notes in a source document](#) for details.

Can contain translatable text

This check box is selected if the segment associated with this markup tag must be translated. This parameter is equivalent to the specification of "TRANSLATEINFO" in [Table 1](#).

On the "Attributes" page you can work with markup attribute definitions. On the left of this page you see the list of attributes that are defined in this markup table. You can select an attribute by clicking it. You can add new attributes (by clicking **Add new attribute**) and delete selected attributes (by clicking **Delete attribute**). On the right of this page you see the definition, respectively properties, of a selected attribute. You can change the properties of an attribute by changing the entry fields and by selecting or clearing the check box.

If you create a new markup attribute, you must at least specify the attribute string and its delimiter, respectively its length.

Attribute string

Contains the string that represents the markup attribute.

End delimiter

Contains one or more characters that end the attribute string. Two or more characters cause OpenTM2 to check for all possible combinations of these characters to determine the end of the attribute. You cannot specify a string as end delimiter. Alternatively, you can specify a length that specifies the end of the attribute string (see parameter **Attribute length**).

Attribute length

Contains the length of the attribute string. Alternatively, you can specify end delimiters that specify the end of the attribute string (see parameter **End delimiter**).

Additional information

Specifies whether specific text is to be ignored when segments are aligned during the creation of an Initial Translation Memory. See "ADDINFO" in [Table 1](#) for a description of possible specifications.

Can contain translatable text

This check box is selected if the segment associated with this markup attribute must be translated. This parameter is equivalent to the specification of "TRANSLATEINFO" in [Table 1](#).

On the "Test" page you can test the markup table. To test a markup table you need to load a test document that contains text that is marked up with tags and attributes. When loaded, the test document is shown in its source form with all markup tags and attributes visible. The test document is analyzed by using the markup table as it is currently defined. As a result of the analysis the contents of the test document is shown with visible distinctions of segments, translatable text, and protected text. You can inspect the test document and decide whether your definitions of markup tags and markup attributes are properly recognized in the test document.

markup tabletesting testingmarkup tables When you test a markup table, you might encounter markup tags or markup attributes in the test document that are not recognized by your markup table. You can add the missing markup tags or markup attributes to your markup table by marking the text section that represents the tag or attribute with the mouse and clicking **Add Selection as tag** or **Add Selection as attribute**. This adds a new markup tag or markup attribute to your markup table. When you click **Refresh** the document is analysed again and the added markup tag or markup attribute should be distinct.

If you encounter markup tags or markup attributes that are improperly recognized during the analysis, you might need to change one or the other property of the affected tag (on the "Tags" page) or attribute (on the "Attributes" page). Thereafter, you can test the effect of the property changes by clicking **Refresh** on the "Test" page.

Load test document

Markup Table Properties window Test page Lets you choose a test document through the operating system's file open dialog, loads the document, and analyses its content by applying the current markup table definitions. The result of the analysis is shown in the text box on the left.

Add Selection as tag

Builds a markup tag from a selected text section in the test document and adds the tag to the markup table. The markup tag is added to the list of tags on the "Tag" page with its default properties.

Add Selection as attribute

Builds a markup attribute from a selected text section in the test document and adds the attribute to the markup table. The markup attribute is added to the list of attributes on the "Attributes" page with its default properties.

Refresh

Restarts the analysis of the loaded test document and shows the result in the text box on the left. Click this button after you added or deleted markup tags or markup attributes, or changed their properties.

Legend

Explains how to interpret markups in the text box. For example, <P> indicates the boundary between segments, and nontranslatable text is marked red.

On all pages of the "Markup Table Properties" window the following buttons are active:

Save

Saves the markup table with its current definitions and leaves the "Markup Table Properties" window.

Cancel

Leaves the "Markup Table Properties" window without saving the markup table.

Protect / Unprotect

markup tablesprotecting protectingmarkup tables Lets you protect or unprotect the markup table by a password against unauthorized changes. If the markup table is unprotected, a **Protect** button is shown. If the markup table is protected, an **Unprotect** button is shown.

Clicking the button invokes a Protect, respectively Unprotect dialog that requests a password. Type a password. Remember the password.

Results

The markup table is now available in **OpenTM2**, you can start to associate it with documents or folders.

Working with language-support files

Language-support files are monolingual morphological data files. With **OpenTM2** they are used to perform:

- **Stem reduction** during dictionary lookup of terms in original documents.

stem reduction

dictionarystem reduction during lookup

When an original document contains the English word **bought**, for example, **OpenTM2** actually looks up the infinitive form of the verb, **buy**, in the dictionary. The English (U.S.) language-support file contains the information necessary to associate **bought** with **buy**.

- **Compound separation** during dictionary lookup of terms in original documents for all Germanic languages.

compound separation

dictionarycompound separation during lookup

When an original document contains the German word **Autoradio** (car radio), **OpenTM2** looks up **Autoradio**. If it does not find this compound, it decomposes it and looks up its parts **Auto** and **Radio**.

- **Text segmentation** into sentences, including recognizing abbreviations in the original documents.

A paragraph of text is divided into sentences during analysis to allow a sentence-by-sentence translation.

- **Spellchecking** of translated documents.

For translations from a given source language, you *must* install the corresponding language-support file.

For translations into a target language, the installation of the language-support file is optional unless you need to check the spelling of the target file.

Overview and terminology

The "Language List" window contains lists of all the language-support files available in **OpenTM2** and their current installation status.

Prerequisites

None.

Calling sequence

Select **Display language list** from the **Utilities** menu.

The "Language List" window is displayed.

Options and parameters

None.

Results

The "Language List" window shows all available languages and their current status. The **State** column indicates whether the language-support file for the respective language is installed (indicated by **active** or **inactive**).

There are several reasons for a language being shown as inactive:

- It is not included in your base package.
- It is included in your base package, but you did not install it.
- It is included in your base package and you installed it, but you subsequently deleted it because you had no need for it at that time.

If you need a language-support file for any of the languages supported by **OpenTM2**, you can install it on top of your current installation.

If you translate into languages for which language support is not available, such as Hungarian, you cannot check the spelling of the translated documents.

What you can do with language-support files

Language-support files can be installed, extended, or deleted.

Deleting language-support files

language-support filesdeleting deletinglanguage-support files You can delete a language-support file when you no longer need it. You may want to do this, for example, if you no longer translate from or into this language.

Prerequisites

None.

Calling sequence

Select:

1. **Display language list** from the **Utilities** menu
2. The language-support file to delete
3. **Delete** from the **File** menu

Options and parameters

Before **OpenTM2** deletes the selected language-support file, you are asked to confirm your request:

- Select **Yes** to delete the language-support file.
- Select **No** to cancel the delete request.

Results

If you select **Yes**, the language-support file is deleted. Otherwise, it remains unchanged. A reference to this source language may still exist in certain folders, dictionaries, and **Translation Memory databases**. Update the respective properties so that the deleted language-support file is no longer referred to.

Extending language-support files

Occasionally, documents contain words with a spelling not recognized by **OpenTM2** although you consider them spelled correctly.

Such words can be added to an *addendum* of the respective language support and are then accepted by the spellchecking function. For more information, see [Spellchecking a document](#).

What you can do for other languages

The list of target languages includes all languages for which language-support files exist, and certain other languages that are supported by Windows^(R). If you want to translate into other languages, select **Other Languages** from the list of languages.

Working with the samples

OpenTM2 comes with sample translation material that you can use for practising. The samples are contained in folders to help you become familiar with the concepts of file organization and navigation in **OpenTM2**.

Overview and terminology

The following table lists the sample folders you can select when installing OpenTM2.

Folder	Format	Original	Translation
samplami	Ami Pro	English (U.S.)	Spanish
samplprt	OS/2 ^(R) resource file	English (U.S.)	German (national)
samplerf	RTF	English (U.S.)	Spanish
samplewp	WordPerfect	English (U.S.)	Spanish
samplew4w	Word 2.0	English (U.S.)	German (national)
samplehtml1	HTML	English (U.S.)	German (national)
sample2	OS/2 ^(R) program file	English (U.S.)	German (national)
samplehtml3	HTML	English (U.S.)	Italian
samplehtml4	HTML	English (U.S.)	Spanish
samplehtml5	HTML	English (U.S.)	French (national)
samplehtml6	HTML	German (national)	English (U.S.)
samplehtml7	HTML	English (U.S.)	Japanese
samplehtml8	HTML	English (U.S.)	Russian
samplehtml9	HTML	English (U.S.)	Arabic
samplehtml10	HTML	English (U.S.)	Polish

Each sample folder contains several documents, a dictionary, and a Translation Memory specially prepared for demonstration purposes. The `samplehtml1` folder, for example, contains:

- Two small documents called `device.html` and `trans.html`
- A small English (U.S.) to German (national) dictionary called `sample1`
- A small **Translation Memory** `samplehtml1`

The text of the sample documents is the same for all formats, except:

- SAMPLE2 contains OS/2^(R) program files
- SAMPLPRT contains a sample dictionary for printing

The document `device.html` is already analyzed, the document `trans.html` is not analyzed.

To install the sample folders, select the **Samples** option during installation. The installation procedure prepares the folders for subsequent import and use.

OpenTM2 performs a morphological stem reduction so that you can look up terms in the original document and check the spelling in translated documents. To use these features with the material in a sample folder, you must first install the appropriate language-support files for the source and target languages. For example, to use folder `samplwp` you must install the English (U.S.) language-support file because the original sample documents are in English. To check which language-support files have been installed, select **Display language list** from the **Utilities** menu of the **OpenTM2** workbench. State **active** tells you that a language has been installed.

To have **OpenTM2** check the spelling of a translated document, you must install the language-support file for the target language. For example, for `SAMPLWP` the Spanish language-support file must be installed.

To study the material in one of the sample folders, import it first. You can then open a document. The **Translation Environment** is displayed and you can start to translate the document. From the "Dictionary" window you can also look up terms in the supplied sample dictionary, and from the "Translation Memory" window you can copy entire sentences.

During translation, sentences you translate are added to the sample **Translation Memory**. To return a sample folder to its initial status, delete the associated dictionary, **Translation Memory**, and folder. Then import the entire folder again. Otherwise, the contents of the initial and updated folders are merged.

To see samples of SGML-based files for data exchange, you can export a sample dictionary or a sample **Translation Memory** in external format.

What you can do with the samples

The following describes a number of tasks related to working with the sample material. They contain short overviews of how to proceed and references to the detailed task descriptions to be found elsewhere in this book.

Deleting a sample folder

If you no longer need any of the sample material, for example, because you are translating other document types or because you are now sufficiently familiar with it, you can delete selected sample folders.

For details on how to delete parts of the samples, see:

- „Deleting a folder“
- „Deleting a document“
- „Deleting a dictionary-entry“
- „Deleting a Translation Memory“

To remove the sample folders from your disk entirely, see [Deleting a folder exported to the `eqlexport` subdirectory](#).

Importing and opening a sample folder and its documents

sample materialimporting sample folders and documents importingsample folders and documents openingsample folders and documents To import a sample folder and to start a translation exercise is described using the folder `samplehtml1` as an example.

Prerequisites

- The samples must have been installed.
- The language-support file for English (U.S.) must have been installed.

Calling sequence

Select the "Folder List" window.

1. Select **Import...** from the **File** menu. The "Import Folder" window is displayed.
2. Select the drive where **OpenTM2** is installed, for example, **C**. The **Folder** list box displays all

- folders that can be imported.
3. Select samplehtml1 from the **Folder** list box.
 4. Click **Details...** This displays the note that is attached to samplehtml1.
 5. Click **OK** to leave the "Folder Details" window.
 6. Click the destination drive where you want to store samplehtml1 .
 7. Click **Import** to begin importing the folder. Several windows are displayed showing the progress of the import procedure.
 8. Click **OK** when you have read the completion message. This takes you back to the **OpenTM2** main window.
 9. Double-click folder samplehtml1 in the "Folder List" window. The "Document List" window is displayed.
 10. Double-click device.html in the "Document List" window. The **Translation Environment** is displayed and the document appears in the Translation window ready for translation.

Results

The folder samplehtml1 is imported into **OpenTM2**. The analyzed document device.html is opened. You can start translating it.

Translation exercise with a sample document

Before documents can be translated, **OpenTM2** analyzes them to prepare them for translation. The device.html documents have already been analyzed. The trans.html documents are analyzed automatically when you open them, or you can explicitly select one of the analysis functions.

After importing and opening a document, you are taken to the **Translation Environment**, where the following windows are displayed:

- The "Translation" window
- The "Translation Memory" window
- The "Dictionary" window

The document device.html contains one chapter of a complete book. Other document files of the book have been translated already, that is, a Translation Memory exists for them.

When you translate a sample document, remember that you can stop at any point in the session by:

- Double-clicking the system menu icon of the **Translation Environment**
- Selecting **Close** from the system menu

To copy a proposal from the **Translation Memory** into the "Translation" window, press and hold down the Control key and type the number of the required proposal.

To copy a proposal from the "Dictionary" window into the "Translation" window, press and hold down the Control key and type the letter of the required proposal.

You can edit the translation document file in either insert or replace mode.

If the suggested translation contains special characters that are not available on your keyboard, ignore them or type other characters instead.

Creating reports

OpenTM2 enables you to collect information about the effort, state, and history of your translations. The collected information can be displayed in different reports and used as a basis for calculating translation costs.

Overview and terminology

OpenTM2 collects information during the following events, called *process tasks*:

- Import of a folder or document
- Export of a folder or document
- Change of folder or document properties
- Analysis of a document
- Automatic substitution during analysis
- Saving of a document
- Deletion of a folder or document

The records containing the information and the result of the collection depend on the process task and are stored in a logging file in compressed form. There is one logging file per folder, the *history log file*, stored as HISTLOG.DAT in the PROPERTY directory of the folder. New records are added

at the end of the history log file.

The following table shows what happens with the history log file during the various process tasks.

Process Task	Action
Importing a folder	<ul style="list-style-type: none"> The imported history log file is merged with the existing one The entries are sorted by date Duplicate entries are stored only once An import record is added to the history log file
Importing a document (external format)	<ul style="list-style-type: none"> An import record is added to the history log file
Importing a document (internal format)	<ul style="list-style-type: none"> The history log file of the imported document is merged with the folder history log file An import record is added to the history log file If a new target document replaces an existing one, an additional record is added containing the results of the target document
Exporting a folder	<ul style="list-style-type: none"> The history log file is exported together with the folder data If only selected documents are exported, only that part of the history log file belonging to these documents is exported
Exporting a document (external or internal format)	<ul style="list-style-type: none"> An export record is added to the history log file
Changing folder or document properties	<ul style="list-style-type: none"> A record containing the new settings is added to the history log file
Analyzing a document	<ul style="list-style-type: none"> An analysis record is added to the history log file If automatic substitution is selected, an additional record containing the collected information is added to the history log file
Saving a document	<ul style="list-style-type: none"> A save record containing the summary of the collected information is added to the history log file
Deleting a folder	<ul style="list-style-type: none"> The history log file is deleted
Deleting a document	<ul style="list-style-type: none"> A deleted record is added to the history log file

During editing and analyzing, information is collected on the following:

- The source and target words
- The quality of the best proposal:
 - Null if an exact proposal exists
 - The number of source words if no proposal exists
 - The number of different words if a fuzzy proposal exists
- The type of the best proposal
- The type of the copied proposal
- Whether a segment is translated using automatic substitution during analysis or while using the editor

The collected information is kept in the segmented target file and deleted when the source file is analyzed again. Each time a document is saved during translation, the information collected is stored in a record that is added to the history log file. If more than one proposal exists for a specific word, only the best proposal is counted and assigned to the appropriate class, independent of your selection. Only those proposals are counted that existed when the segment was translated for the first time. An exact proposal is assigned to the column "Analyze Auto", "Analyze Edit", or "Edit Exact".

The number of source words is counted when a segment is activated, which means that segments not yet translated always have zero source words. The number of target words is updated each time a segment is saved and when the segment is autosubstituted. For the current segment, the number of target words is zero if not yet translated. If a source segment is copied into a target document, the number of target words is set to zero. The following rules apply for counting source and target segments:

- The same counting facility is used as for counting the number of words in documents.
- Punctuation and NOLOOKUP tokens are not counted.

- Inline tags are not counted. For inline tags with attributes, only the translatable information is counted.

The number of modified words, which is the number of words differing in the current source and the source of proposal, is counted using an LCS algorithm. Only the best proposal is counted. The modified words are counted when the segment is saved for the first time. It is not counted when the segment is activated, but not translated. The count is not changed when the segment is translated again. The information on which proposal has been chosen is not saved in the history log file. The current segment belongs to the count of segments not translated. If two segments are joined, they are counted as one.

The reports created from the collected information can be either displayed in a window or stored in a file. Each report is attached to the specified folder and can be created for the whole folder or selected documents in the folder. You can create the following types of report:

- The **History Report** keeps track of the most important process tasks performed by the translator.
- The **Counting Report** enables you to roughly calculate the effort of a translator. You cannot customize this report to fit your needs.
- The **Calculating Report** enables you to exactly calculate the effort of a translator. You can customize this report to fit your needs.
- The **Preanalysis Report** calculates the effort before a translation starts by checking the **Translation Memory databases** for exact and fuzzy matches.
- The **Redundancy Report** analyzes redundancies in a folder or document to calculate the effort required by a translation. Use this report to calculate the translation costs before a project starts.
- The **Redundant Segment List** lists the 100 most frequently used segments in a folder.

The following table gives an overview of when a report should be created.

When to create	Before translation	During translation	After translation
Preanalysis Report	x		
Redundancy Report	x		
Redundant Segment List	x		
Calculating Report	(x)	x	x
History Report		x	x
Counting Report		x	x

The Calculating Report is the most important report for all parties involved in a translation project. All other reports help you manage your translations in a more effective way.

To create a report or list, a folder containing at least one document must exist.

Select:

1. The folder from the "Folder List" window or a document from the "Document List" window.
2. Create Counting Report... from the **Utilities** menu.

The "Create Counting Report" window (see [Figure 123](#)) is displayed.

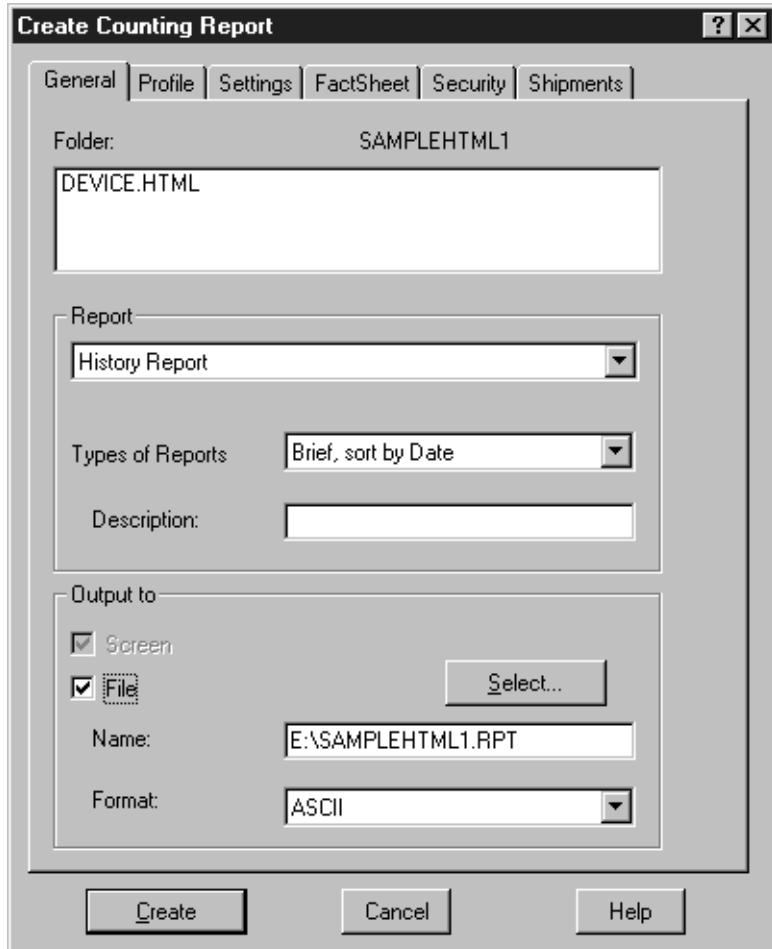


Figure 123. Create Counting Report window

The options and parameters available depend on the report you want to create. The following sections describe the individual reports.

Creating a History Report

reportsHistory Report History Report The History Report contains a history of the process tasks with regard to the selected folder or document. It keeps track of the most important tasks performed by the translator. It is available in different views depending on the level of complexity of the process tasks to be tracked. It should be created if problems with word-count results or consistency arise and when a translation project is finished.

If you want to create a report for an entire folder, all documents that it ever contained are listed.

Calling sequence

Select **History Report** from the **Report** field.

The following table shows which tabbed pages must be filled in to produce a History Report.

Table 2. Overview of the tabbed pages to be filled in for a History Report

Tabbed page	Purpose	Required?	Described in...
"General"	To define the basics required for a report	Yes	"General page"
"Profile"	To save or load the report layout defined on the "Settings" tabbed page	No	"Profile page"
"Settings"	To define the report layout	No	
"FactSheet"	To set the factors for cost calculation	No	"Setting the factors for cost calculation"
"Security"	To select special options	Optional	"Security page"
"Shipments"	To create a report on specific shipments	No	"Shipments page"

Options and parameters

On the "General" page (see [Figure 123](#)), you can choose or specify the following options and parameters:

Folder

This box contains the documents contained in the selected folder or the document that you selected from the **Document List**.

Report types

Choose one of the following History Report types:

Brief, sort by date

To get a brief report that is sorted by the creation date and time of the documents and contains the following information:

- A unique record number within the history log file
- The name of the documents for which you create a history report
- The creation date and time of the documents
- The process tasks for each document listed in the report

Brief, sort by document

To get a brief report where the information is sorted by document name.

Detail

To get a detailed report of the process tasks performed on the selected documents, such as the **Translation Memory** and markup language used and the number of matches found. The information is sorted by the unique record number.

Version

To get a report on the **OpenTM2** versions used.

Description

To specify additional information on the report that will display in the header of the report. You can type up to 80 characters.

Output to Screen

To display the report in a window.

Output to File

To store the report in a specific file for which you can specify or select the name and directory and select the file format.

Click the "Security" tab to display the "Security" tabbed page (see [Figure 124](#)).



Figure 124. Create Counting Report window — Security page

On the "Security" page, you can choose or specify the following options and parameters:

List of documents

Select this option if you want to add, to the end of the report, a list of all documents contained in this report. This is useful if your documents have long names because the report abbreviates file names to 10 characters. The list at the end of a report, however, contains the full document names.

Click **Create** to start the creation of the History Report.

Results

The History Report is created according to your specifications. The following examples show a brief History Report, sorted by date, on one document and an excerpt of a detailed History Report, also on one document.

Example 1:

Task Id	Document	Date Time	Process task
1	DEVICE.SCR	08.09.1999 16:11:08	Document import
4	DEVICE.SCR	08.09.1999 16:25:31	Save of document expanded form
6	DEVICE.SCR	09.09.1999 16:10:47	Document analysis
7	DEVICE.SCR	09.09.1999 16:10:47	...with automatic substitution
8	DEVICE.SCR	09.09.1999 16:11:16	Document analysis
9	DEVICE.SCR	09.09.1999 16:11:18	...with automatic substitution
10	DEVICE.SCR	09.09.1999 16:11:39	Document analysis
11	DEVICE.SCR	09.09.1999 16:11:39	...with automatic substitution

Figure 125. Brief History Report, sorted by date

Example 2:

1	Document import
Date	08.09.1999 16:11:08
Document	DEVICE.SCR
Format	Folder
Folder	SAMPLE1
Source replaced	No
Source replaced	No

4	Save of document expanded form
Date	08.09.1999 16:25:31
Document	DEVICE.SCR

		Segments	Source	Modified	Target
			Words	Words	Words
None	1..4	0	0	0	0
Matches	5..14	1	5	5	6
	>=15	0	0	0	0
Not	1..4	2	7	0	0
Translated	5..14	10	70	0	0
	>=15	4	84	0	0

Figure 126. Detailed History Report

Creating a Counting Report

reportsCounting Report Counting Report The Counting Report contains tables that enable you to roughly calculate the translator's effort. You cannot change the tables to meet your requirements.

Calling sequence

Select **Counting Report** from the **Report** field.

The following table shows which tabbed pages must be filled in to produce a Counting Report.

Table 3. Overview of the tabbed pages to be filled in for a Counting Report

Tabbed page	Purpose	Required?	Described in...
"General"	To define the basics required for a report	Yes	"General page"
"Profile"	To save or load the report layout defined on the Settings tabbed page	No	"Profile page"
"Settings"	To define the report layout	No	
"FactSheet"	To set the factors for cost calculation	No	"Setting the factors for cost calculation"
"Security"	To select special options	Optional	"Security page"
"Shipments"	To create a report on specific shipments	No	"Shipments page"

Options and parameters

On the "General" page (see [Figure 123](#)), you can choose or specify the following options and parameters:

Folder

This box contains the documents contained in the selected folder or the document that you selected from the **Document List**.

Report types

Choose one of the following Counting Report types:

Base list

To get a redundant segment list containing only the 99 most used segments with a frequency of 3 or above.

Detailed list

To get a redundant segment list containing all redundant segments.

Description

To specify additional information on the report that will display in the header of the report. You can type up to 80 characters.

Output to Screen

To display the report in a window.

Output to File

To store the report in a specific file for which you can specify or select the name and directory and select the file format.

Click **Create** to start the creation of the Counting Report.

Results

The Counting Report is created according to your specifications. The following example shows a Counting Report on the contents of a folder, with a summary at its end. For an explanation of the individual rows, refer to [The report layout](#). When interpreting the results note that the criteria for fuzzy matches can be customized, as described in [#unresolvedid/viewsysprop](#). This also influences the "No match" counter. **fuzzy match overlap ratioimpact on counting report**

Document		DEVICE.SCR			
Date		09.09.1999 16:11:39			
Last process task		...with automatic substitution			
		Segments	Source Words	Modified Words	Target Words
Analysis	1..4	0	0	0	0
Autosubst	5..14	0	0	0	0
	>=15	0	0	0	0
Edit	1..4	0	0	0	0
Autosubst	5..14	0	0	0	0
	>=15	0	0	0	0
Exact	1..4	0	0	0	0
Matches	5..14	0	0	0	0
	>=15	0	0	0	0
Replace	1..4	0	0	0	0
Matches	5..14	0	0	0	0
	>=15	0	0	0	0
Fuzzy	1..4	0	0	0	0
Matches	5..14	0	0	0	0
	>=15	0	0	0	0
Machine	1..4	0	0	0	0
Matches	5..14	0	0	0	0
	>=15	0	0	0	0
None	1..4	0	0	0	0
Matches	5..14	1	5	5	6
	>=15	0	0	0	0
Not	1..4	1	4	0	0
Translated	5..14	5	31	0	0
	>=15	0	0	0	0
Summary	1..4	0	0	0	0
	5..14	1	5	5	6
	>=15	0	0	0	0

Figure 127. Counting Report with totals — Table on first document in folder

Selected documents of folder SAMPLE1					
Summary generated at 01.12.1999 14:56:19					
		Segments	Source Words	Modified Words	Target Words
			Words	Words	Words
Analysis	1..4	0	0	0	0
Autosubst	5..14	0	0	0	0
	>=15	0	0	0	0
Edit	1..4	0	0	0	0
Autosubst	5..14	0	0	0	0
	>=15	0	0	0	0
Exact	1..4	0	0	0	0
Matches	5..14	1	11	0	11
	>=15	0	0	0	0
Replace	1..4	0	0	0	0
Matches	5..14	0	0	0	0
	>=15	0	0	0	0
Fuzzy	1..4	0	0	0	0
Matches	5..14	0	0	0	0
	>=15	0	0	0	0
Machine	1..4	0	0	0	0
Matches	5..14	0	0	0	0
	>=15	0	0	0	0
None	1..4	0	0	0	0
Matches	5..14	1	5	5	6
	>=15	0	0	0	0
Not	1..4	3	8	0	0
Translated	5..14	14	110	1	0
	>=15	2	47	0	0
Summary	1..4	0	0	0	0
	5..14	2	16	5	17
	>=15	0	0	0	0

Figure 128. Counting Report with totals — Summary table

Creating a Calculating Report

reportsCalculating Report Calculating Report The Calculating Report is the most important report for all parties involved in a translation project. It enables you to exactly calculate the effort for a translation project. It contains detailed tables, a summary, a fact sheet, or a combination of the three, on the contents of a folder or on one or more documents, depending on your specifications. Create such a report before a translation project starts, during a translation project, or after it is finished.

Calling sequence

Select **Calculating Report** from the **Report** field.

The following table shows which tabbed pages must be filled in to produce a Calculating Report.

Table 4. Overview of the tabbed pages to be filled in for a Calculating Report

Tabbed page	Purpose	Required?	Described in...
"General"	To define the basics required for a report	Yes	"General page"
"Profile"	To save or load the report layout defined on the Settings tabbed page	Optional	"Profile page"
"Settings"	To define the report layout	Optional	
"FactSheet"	To set the factors for cost calculation	Optional	"Setting the factors for cost calculation"
"Security"	To select special options	Optional	"Security page"
"Shipments"	To create a report on specific shipments	Optional	"Shipments page"

Options and parameters

On the "General" page (see [Figure 123](#)), you can choose or specify the following options and parameters:

Folder

This box contains the documents contained in the selected folder or the document that you selected from the **Document List**.

Report types

Choose one of the following Calculating Report types:

Base

To get a detailed overview, for each selected document, of the activities performed by **OpenTM2** and by the translator to complete a translation. The translator's effort can be calculated based on these activities.

Summary

To get a summary row for each selected document, and a summary row for all documents.

Fact sheet

To get a complete overview of the costs of a translation project using different complexity and pay factors.

Base & Summary & Fact Sheet

To get a report containing all the information previously described.

Base & Summary

To get the detailed overview, a summary row for each selected document, and a summary row for all documents.

Summary & Fact Sheet

To get a report containing both the summary and the fact sheet. This report is designed for the project coordinator.

Base & Summary & Fact Sheet

To get a report containing all the information previously described.

Description

To specify additional information on the report that will display in the header of the report. You can type up to 80 characters.

Output to Screen

To display the report in a window.

Output to File

To store the report in a specific file for which you can specify or select the name and directory and select the file format.

Click the "Profile" tab to display the "Profile" page (see [Figure 129](#)).

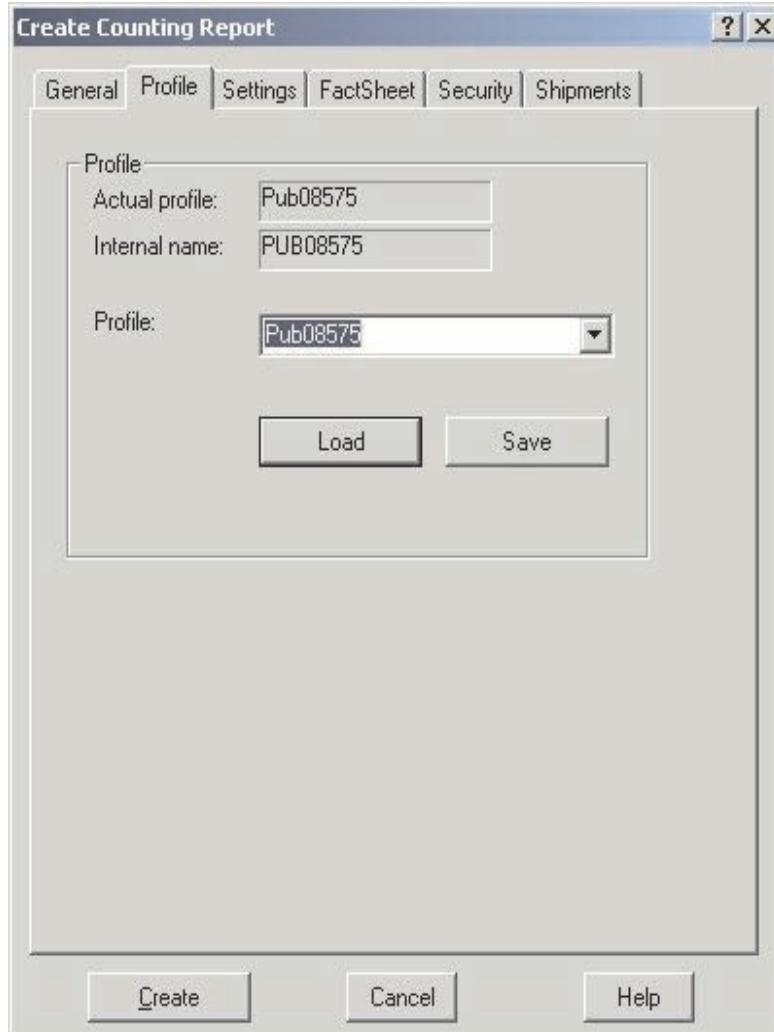


Figure 129. Create Counting Report window — Profile page
[\[PIC\]](#)Figure shows Create Counting Report window — Profile page

On the "Profile" page, you can choose the following option:

Actual profile

This display-only field shows the name of the loaded profile. If no profile has been loaded "- none —" is shown.

Internal name

This display-only field shows the internal name of the loaded profile. Note: only official approved profiles have an internal profile name which is normally the same as name of the profile.

Profile

Type a name for the settings that you specified on the "Settings" tabbed page and then click **Save** to save them. Or select or specify the name of existing settings that you want to use and then click **Load**.

Click the "Shipments" tab to display the "Shipments" page (see [Figure 130](#)).

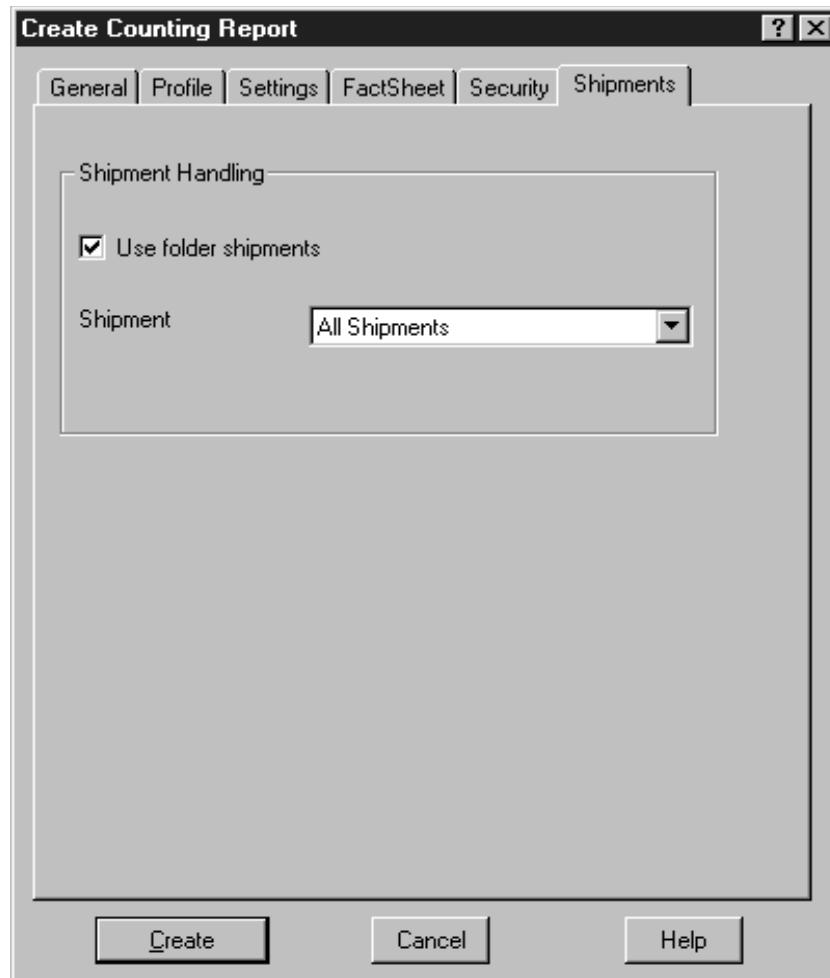


Figure 130. Create Counting Report window — Shipments page
[\[PIC\]](#)Figure shows Create Counting Report window — Shipments page

On the "Shipments" page, you can choose or specify the following options and parameters:

Use folder shipments

Select this option if you want to use the original shipment number of the folder.

Shipment

You can select one of the following:

All shipments

To get a report on all shipments

Single shipment

To get a report on all shipments, ordered by the shipment number.

Shipment number

The shipment number of the folder for which you want to create the report. The shipment number of a folder only appears in this field if you defined it during the creation of the folder (see [Creating a folder](#)) or when changing the properties of the folder (see [Changing the properties of a folder](#)).

Click **Create** to start the creation of the Calculating Report.

Results

The Calculating Report is created according to your specifications. The following example shows a Calculating Report containing the overview, summary, and fact sheet of two documents. For an explanation of the individual columns, refer to [The report layout](#).

Document		DEVICE.SCR												
Folder		C:\EQF\SAMPLE1.F00												
		Source Words												
Doc Id	Document	Shipm	Catego.	Auto	Post	Edit	Exact	Replace	50-70%	70-90%	>90%	Mach.	None	Not
1	DEVICE.SCR	1	1..4		3									4
			5..14		44								5	31
			>= 15		84									

Document		TRANS.DOC												
Folder		C:\EQF\SAMPLE1.F00												
		Source Words												
Doc Id	Document	Shipm	Catego.	Auto	Post	Edit	Exact	Replace	50-70%	70-90%	>90%	Mach.	None	Not
2	TRANS.DOC	1	1..4											4
			5..14										11	79
			>= 15											47

Figure 131. Calculating Report — Base

Document		Summary												
Folder		C:\EQF\SAMPLE1.F00												
		Source Words												
Doc Id	Document	Shipm	Catego.	Auto	Post	Edit	Exact	Replace	50-70%	70-90%	>90%	Mach.	None	Not
1	DEVICE.SCR	1	1..4		3									4
			5..14		44								5	31
			>= 15		84									
2	TRANS.DOC	1	1..4											4
			5..14										11	79
			>= 15											47
	Sum	1	1..4		3									8
			5..14		44								16	110
			>= 15		84									47
		1	Sum		131								16	165

Figure 132. Calculating Report — Summary

Document Folder		Final Fact Sheet C:\EQF\SAMPLE\FOO Source Words												
Doc Id	Document	Number		Analyze			Edit		Fuzzy			Mach.	None	Not
		Shipm	Catego.	Auto	Post	Edit	Exact	Replace	50-70%	70-90%	>90%	Matches	Matches	Transl
	Actual Words	1	1..4		3									8
			5..14		44								16	110
			>=15		84									47
	Complexity		1..4	1	1	1	1	1	1	1	1	1	1	1
	Factor		5..14	1	1	1	1	1	1	1	1	1	1	1
			>=15	1	1	1	1	1	1	1	1	1	1	1
	Complexity		1..4		3									
	Factor *		5..14		44								16	
	Actual Words		>=15		84									
	Pay		1..4	1	1	1	1	1	1	1	1	1	1	1
	Factor		5..14	1	1	1	1	1	1	1	1	1	1	1
			>=15	1	1	1	1	1	1	1	1	1	1	1
	Pay		1..4		3									
	Factor *		5..14		44								16	
	Compl. Word		>=15		84									
	Payable Words		Sum		131								16	

Final Summary	
Payable Words	147.00
Local Currency	0.00 ARP
Total Pay	0.00 ARP

Figure 133. Calculating Report — Fact Sheet

Note:

If you create this report before a translation starts, only the “Analyze Auto” and “Not Translated” columns contain figures. During and after a translation, all other columns can also contain figures. After a translation is complete, the “Not Translated” column should show “0”.

Creating a Preanalysis Report

reportsPreanalysis Report Preanalysis Report The Preanalysis Report is used to calculate the translator’s effort on a “what-if” basis. It simulates a translation using exact and fuzzy matches from all **Translation Memory databases** belonging to the folder or documents. Create this report before work on a translation project starts.

Prerequisites

To prepare a folder or document for this report type, analyze it with the “Count Translation Memory match information” option selected. See [Analyzing documents using Translation Memory databases](#) for more information.

Calling sequence

Select **Preanalysis Report** from the **Report** field. The following table shows which tabbed pages must be filled in to produce a Preanalysis Report.

Table 5. Overview of the tabbed pages to be filled in for a Preanalysis Report

Tabbed page	Purpose	Required?	Described in..
“General”	To define the basics required for a report	Yes	“General page”
“Profile”	To save or load the report layout defined on the Settings tabbed page	Optional	“Profile Page”
“Settings”	To define the report layout	Optional	
“FactSheet”	To set the factors for cost calculation	Optional	“Setting the factors for count calculation”
“Security”	To select special options	Required?	“Security page”
“Shipments”	To create a report on	Optional	“Shipments page”

Tabbed page	Purpose	Required?	Described in...
	specific shipments		

Options and parameters

On the "General" page (see [Figure 1](#)), you can choose or specify the following options and parameters:

Folder

This box contains the documents contained in the selected folder or the document that you selected from the **Document List**.

Report types

Choose one of the following Preanalysis Report types:

Base

To get a detailed overview, for each selected document, of the activities performed by **OpenTM2** and by the translator to complete a translation. The translator's effort can be calculated based on these activities.

Summary

To get a summary row for each selected document, and a summary row for all documents.

Fact sheet

To get a complete overview of the costs of a translation project using different complexity and pay factors.

Base & Summary & Fact Sheet

To get a report containing all the information previously described.

Base & Summary

To get the detailed overview, a summary row for each selected document, and a summary row for all documents.

Summary & Fact Sheet

To get a report containing both the summary and the fact sheet. This report is designed for the project coordinator.

Description

To specify additional information on the report that will display in the header of the report. You can type up to 80 characters.

Output to Screen

To display the report in a window.

Output to File

To store the report in a specific file for which you can specify or select the name and directory and select the file format.

Click **Create** to start the creation of the Preanalysis Report.

Results

The Preanalysis Report is created according to your specifications. The following example shows you a Preanalysis Report containing the overview, summary, and fact sheet of two documents. For an explanation of the individual columns, refer to [The report layout](#).

Document			DEVICE.SCR									
Folder			C:\EQF\SAMPLE1\FOO									
Memory's			SAMPLE1 ,									
Source Words												
Doc Id	Document	Catego.	Analyze	Edit		Fuzzy			Mach.	None	Not	
			Auto	Exact	Replace	50-70%	70-90%	>90%	Matches	Matches	Transl	
1	DEVICE.SCR	1..4	3	4								
		5..14	44	5					5		21	
		>= 15	84									

Document			TRANS.DOC									
Folder			C:\EQF\SAMPLE1\FOO									
Memory's			SAMPLE1 ,									
Source Words												
Doc Id	Document	Catego.	Analyze	Edit		Fuzzy			Mach.	None	Not	
			Auto	Exact	Replace	50-70%	70-90%	>90%	Matches	Matches	Transl	
2	TRANS.DOC	1..4	4									
		5..14	60						19		11	
		>= 15	47									

Figure 134. Preanalysis Report — Base

Document			Summary									
Folder			C:\EQF\SAMPLE1\FOO									
Memory's			SAMPLE1 ,									
Source Words												
Doc Id	Document	Catego.	Analyze	Edit		Fuzzy			Mach.	None	Not	
			Auto	Exact	Replace	50-70%	70-90%	>90%	Matches	Matches	Transl	
1	DEVICE.SCR	1..4	3	4								
		5..14	44	5					5		21	
		>= 15	84									
2	TRANS.DOC	1..4	4									
		5..14	60						19		11	
		>= 15	47									
	Sum	1..4	7	4								
		5..14	104	5					24		32	
		>= 15	131									
	Sum	242	9						24		32	

Figure 135. Preanalysis Report — Summary

Document Folder Memory's		Final Fact Sheet C:\EQF\SAMPLE1.FOO SAMPLE1 , Source Words										
Doc Id	Document	Catego.	Analyze	Edit			Fuzzy			Mach.	None	Not
			Auto	Exact	Replace		50-70%	70-90%	>90%	Matches	Matches	Transl
	Actual Words	1..4	7	4								
		5..14	104	5					24		32	
		>= 15	131									
	Complexity	1..4	1	1	1		1	1	1	1	1	
	Factor	5..14	1	1	1		1	1	1	1	1	
		>= 15	1	1	1		1	1	1	1	1	
	Complexity	1..4	7	4								
	Factor *	5..14	104	5					24		32	
	Actual Words	>= 15	131									
	Pay	1..4	1	1	1		1	1	1	1	1	
	Factor	5..14	1	1	1		1	1	1	1	1	
		>= 15	1	1	1		1	1	1	1	1	
	Pay	1..4	7	4								
	Factor *	5..14	104	5					24		32	
	Compl. Word	>= 15	131									
	Payable Words	Sum	242	9					24		32	

Final Summary	
Payable Words	307.00
Local Currency	0.00 ARP
Total Pay	0.00 ARP

Figure 136. Preanalysis Report — Fact Sheet

Creating a Redundancy Report

reportsRedundancy Report Redundancy Report The Redundancy Report is used to calculate the translator's effort on a "what-if" basis. It simulates a translation using exact and fuzzy matches from all **Translation Memory databases** belonging to the folder or documents. In addition, it counts the redundant sentences, that is, the sentences that appear more than once, within each document (inner-document redundancies) and across all documents in the folder (cross-document redundancies). Create this report before work on a translation project starts.

Prerequisites

To prepare a folder or document for this report type, analyze it with the "Count Translation Memory match information" and "Prepare Redundancy Report" options selected. See [Analyzing documents using Translation Memory databases](#) for more information.

Calling sequence

Select **Redundancy Report** from the **Report** field.

The following table shows which tabbed pages must be filled in to produce a Redundancy Report.

Table 6. Overview of the tabbed pages to be filled in for a Redundancy Report

Tabbed page	Purpose	Required?	Described in...
"General"	To define the basics required for a report	Yes	"General page"
"Profile"	To save or load the report layout defined on the Settings tabbed page	Optional	"Profile page"
"Settings"	To define the report layout	Optional	
"FactSheet"	To set the factors for cost calculation	Optional	"Setting the factors for count calculation"
"Security"	To select special options	Optional	"Security page"
"Shipments"	To create a report on specific shipments	Optional	"Shipments page"

Options and parameters

On the "General" page (see [Figure 1](#)), you can choose or specify the following options and parameters:

Folder

This box contains the documents contained in the selected folder or the document that you selected from the **Document List**.

Report types

Choose one of the following Redundancy Report types:

Base

To get a detailed overview, for each selected document, of the activities performed by **OpenTM2** and by the translator to complete a translation. The translator's effort can be calculated based on these activities.

Summary

To get a summary row for each selected document, and a summary row for all documents.

Fact sheet

To get a complete overview of costs of a translation using different complexity and pay factors.

Base & Summary & Fact Sheet

To get a report containing all the information previously described.

Base & Summary

To get the detailed overview, a summary row for each selected document, and a summary row for all documents.

Summary & Fact Sheet

To get a report containing both the summary and the fact sheet. This report is designed for the project coordinator.

Description

To specify additional information on the report that will display in the header of the report. You can type up to 80 characters.

Output to Screen

To display the report in a window.

Output to File

To store the report in a specific file for which you can specify or select the name and directory and select the file format.

Click **Create** to start the creation of the Redundancy Report.

Results

The Redundancy Report is created according to your specifications. The following example shows you a Redundancy Report containing the overview, summary, and fact sheet of two documents. For an explanation of the individual columns, refer to [The report layout](#).

Document		DEVICE.SCR													
Folder		C:\EQF\SAMPLE1.FOO													
Source Words		Inner Document Redundancies													
Doc Id	Document	Number	Shipm	Catego.	Analyze	Post	Edit	Exact	Replace	50-70%	70-90%	>90%	Mach.	None	Not
1	DEVICE.SCR	1	1..4	3				4					Matches	Matches	Transl
			5..14	44				5				5		21	
			>= 15	84											

Document		TRANS.DOC													
Folder		C:\EQF\SAMPLE1.FOO													
Source Words		Inner Document Redundancies													
Doc Id	Document	Number	Shipm	Catego.	Analyze	Post	Edit	Exact	Replace	50-70%	70-90%	>90%	Mach.	None	Not
2	TRANS.DOC	1	1..4	4									Matches	Matches	Transl
			5..14	60								19		11	
			>= 15	47											

Document		Redundancies													
Folder		C:\EQF\SAMPLE1.FOO													
Source Words		Cross Document Redundancies													
Doc Id	Document	Number	Shipm	Catego.	Analyze	Post	Edit	Exact	Replace	50-70%	70-90%	>90%	Mach.	None	Not
3	Redundancies	1	1..4										Matches	Matches	Transl
			5..14					32					-		
			>= 15												

Figure 137. Redundancy Report — Base

Document		Summary													
Folder		C:\EQF\SAMPLE1.FOO													
Source Words		All Redundancies													
Doc Id	Document	Number	Shipm	Catego.	Analyze	Post	Edit	Exact	Replace	50-70%	70-90%	>90%	Mach.	None	Not
1	DEVICE.SCR	1	1..4	3				4					Matches	Matches	Transl
			5..14	44				5				5		21	
			>= 15	84											
2	TRANS.DOC	1	1..4	4									19		11
			5..14	60											
			>= 15	47											
3	Redundancies	1	1..4										-		
			5..14				32								
			>= 15												
	Sum	1	1..4	7				4							
			5..14	104				37				24			
			>= 15	131											
		1	Sum	242				41				24			

Figure 138. Redundancy Report — Summary

Document		Final Fact Sheet											
Folder		C:\EQF\SAMPLE\FOO											
Source Words		Inner Document Redundancies											
Doc Id	Document	Number		Analyze		Edit		Fuzzy		Mach.	None	Not	
			Shipm	Catego.	Auto	Post	Edit	Exact	Replace	50-70%	70-90%	>90%	Matches
	Actual Words	1	1..4	7				4					
			5..14	104				37				24	
			>= 15	131									
	Complexity	1..4	1	1	1	1	1	1	1	1	1	1	1
	Factor	5..14	1	1	1	1	1	1	1	1	1	1	1
		>= 15	1	1	1	1	1	1	1	1	1	1	1
	Complexity	1..4	7				4						
	Factor *	5..14	104				37					24	
	Actual Words	>= 15	131										
	Pay	1..4	1	1	1	1	1	1	1	1	1	1	1
	Factor	5..14	1	1	1	1	1	1	1	1	1	1	1
		>= 15	1	1	1	1	1	1	1	1	1	1	1
	Pay	1..4	7			4							
	Factor *	5..14	104			37						24	
	Compl. Word	>= 15	131										
	Payable Words	Sum	242				41					24	

Final Summary	
Payable Words	307.00
Local Currency	0.00 ARP
Total Pay	0.00 ARP

Figure 139. Redundancy Report — Fact Sheet

Creating a Redundant Segment List

The Redundant Segment List contains the first 100 most frequently used segments in a folder. As part of your translation project preparation, translate the segments listed. You can open the documents containing the segments from the Redundant Segment List by double-clicking the document name.

Prerequisites

To prepare a folder or document for this report type, analyze it with the “Count Translation Memory” match information” and “Prepare Redundant Segment List” options selected. See [Analyzing documents using Translation Memory databases](#) for more information.

Calling sequence

Select **Redundant Segment List** from the **Report** field.

The following table shows which tabbed pages must be filled in to produce a Redundancy Segment List.

Table 7. Overview of the tabbed pages to be filled in for a Redundancy Segment List

Tabbed page	Purpose	Required?	Described in...
“General”	To define the basics required for a report	Yes	“General page”
“Profile”	To save or load the report layout defined on the Settings tabbed page	No	“Profile page”
“Settings”	To define the report layout	No	
“FactSheet”	To set the factors for cost calculation	No	“Setting the factors for count calculation”
“Security”	To select special options	Optional	“Security page”
“Shipments”	To create a report on specific shipments	No	“Shipments page”

To produce a Redundant Segment List, fill in the following tabbed pages:

- "General", described in the following
- Optionally, "Security", described in "Security page"

Options and parameters

On the "General" page (see [Figure 1](#)), you can choose or specify the following options and parameters:

Folder

This box contains the documents contained in the selected folder or the document that you selected from the **Document List**.

Description

To specify additional information on the report that will display in the header of the report. You can type up to 80 characters.

Output to Screen

To display the report in a window.

Output to File

To store the report in a specific file for which you can specify or select the name and directory and select the file format.

Click **Create** to start the creation of the Redundant Segment List.

Results

The Redundant Segment List is created according to your specifications. The following example shows a Redundant Segment List.

```
Entry 0: Frequency=4
[1] C:\EQF\SAMPLE1.FOO\translate.doc : #1
Segment: Do not translate this sentence.

=====
Entry 1: Frequency=3
[1] C:\EQF\SAMPLE1.FOO\translate.doc : #6
Segment: Error!
```

Figure 140. Redundant Segment List

Changing the layout of a report

The Calculating, Preanalysis, and Redundancy Reports have the same layout. However, you can change it according to your needs. The following sections explain the various rows and columns of a report and tell you what you can change.

The report layout

This section describes the individual rows and columns of a Calculating, Preanalysis, or Redundancy Report.

Analyze Auto

Contains the number of exact and exact-exact matches retrieved from a Translation Memory during the analysis of a document for which the manual translation has already been started. The "**Analyze Auto2**" numbers are not used for payment purposes as these numbers contain a mixture of manually translated segments (which have been added to the memory while manually translating the document) and segments which were already in the memory before the manual translation started.

When the document is re-imported OpenTM2 assumes the start of a new shipment. The automatically substituted segments during an analysis following the re-import are when counted in the "**Analyze Auto**" column although the manual translation may have been started already.

Analyze Auto2

Contains the number of exact and exact-exact matches retrieved from a **Translation Memory** during the analysis of a document. To get a result, you must select the option "Substitute exact matches in the documents automatically" on the "General" page of the "Analyze Documents" window (see [Analyzing documents using Translation Memory databases](#) for a description). If more than one exact match is found, it is counted as part of the "Edit Exact" column because the

translator has to decide which match is the correct one.

Only the first analysis of the document is taken into account. If, however, a previously translated document is imported and analyzed again, the report contains both the number of exact matches of the first analysis and the results of the reimported document. The results are shown as belonging to shipment 1 and shipment 2 (column "Number Shipment").

Analyze Post

Has the same contents as, and replaces, the "Analyze Auto" column after the analyzed document is opened for translation.

Analyze Edit

Contains the number of exact matches retrieved from a **Translation Memory** during the translation of the document. To get a result, you must analyze the document without the "Substitute exact matches in the documents automatically" option and then select the option "Automatic substitution" from the **Translate** menu during translation. Only the first analysis of the document is taken into account.

If a previously translated document is imported and analyzed again and then edited with automatic substitution, the report contains the number of exact matches of both the first and the second edit. The results are shown as belonging to shipment 1 and shipment 2 (column "Number Shipment").

Edit Auto

Contains the number of exact and exact-exact matches retrieved from a Translation Memory during the "**Automatic Substitution**" from within the Translation Environment.

Edit Exact

Contains the number of exact matches copied from a **Translation Memory** during the translation of the document, using the $\text{Ctrl}+n$ key (where n is the number of the exact-match proposal). In addition, it includes the number of exact matches where the translator had to choose between several matches offered. It also includes the number of inner-document and cross-document redundancies produced by the Redundancy Report.

If a previously translated document is imported, analyzed, and edited again, the report contains the number of exact matches of both the first and the second edit. The results are shown as belonging to shipment 1 and shipment 2 (column "Number Shipment").

If you selected the "Use existing proposals" option on the "Settings" page, this column shows all exact matches offered by the **Translation Memory** regardless of whether the translator accepted and copied them. The number of exact matches not accepted by the translator is reflected in the Calculating Report when you choose "Include statistics" and "Advanced statistics" on the "Settings" page.

Edit Replace

Contains the number of fuzzy replacement matches copied from an existing **Translation Memory** during the translation of the document, using the $\text{Ctrl}+n$ key (where n is the number of the fuzzy-replacement-match proposal).

If a previously translated document is imported, analyzed, and edited again, the fuzzy replacement matches become exact matches. **OpenTM2** keeps the original number of fuzzy replacement matches and adds the number of exact matches to the "Analyze Auto" column as shipment 2.

If you selected the "Use existing proposals" option on the "Settings" page, this column shows all fuzzy replacement matches offered by the **Translation Memory** regardless of whether the translator accepted and copied them. The number of fuzzy replacement matches not accepted by the translator is reflected in the Calculating Report when you choose "Include statistics" and "Advanced statistics" on the "Settings" page.

Fuzzy Matches

Contains the number of fuzzy matches copied from an existing **Translation Memory** during the translation of the document, using the $\text{Ctrl}+n$ key (where n is the number of the fuzzy-match proposal). The fuzzy matches are broken down to matches that fit 50 to 70%, 70 to 90%, and over 90%.

If a previously translated document is imported, analyzed, and edited again, the fuzzy matches become exact matches. **OpenTM2** keeps the original number of fuzzy matches and adds the number of exact matches to the "Analyze Auto" column as shipment 2.

If you selected the "Use existing proposals" option on the "Settings" page, this column shows all fuzzy matches offered by the **Translation Memory** regardless of whether the translator accepted and copied them. The number of fuzzy matches not accepted by the translator is reflected in the Calculating Report when you choose "Include statistics" and "Advanced statistics" on the "Settings" page.

Machine Matches

Contains the number of exact matches copied from an **Initial Translation Memory** during the translation of the document, using the **Ctrl+n** key (where *n* is the number of the machine-match proposal).

If a previously translated document is imported, analyzed, and edited again, the machine matches become exact matches. **OpenTM2** keeps the original number of machine matches and adds the number of exact matches to the “Analyze Auto” column as shipment 2.

If you selected the “Use existing proposals” option on the “Settings” page, this column shows all machine matches offered by the **Translation Memory** regardless of whether the translator accepted and copied them. The number of machine matches not accepted by the translator is reflected in the Calculating Report when you choose “Include statistics” and “Advanced statistics” on the “Settings” page.

Non Matches

Contains the number of segments translated manually. This number remains the same even if you analyze the translated document again and translate it using the “Automatic substitution” option from the **Translate** menu.

If a previously translated document is imported, analyzed, and edited again, the non-matches become exact matches. **OpenTM2** keeps the original number of non-matches and adds the number of exact matches to the “Analyze Auto” column as shipment 2.

Not Translated

Contains the number of segments not yet translated because the translation has not started yet or was interrupted. If the partly translated document is imported and analyzed again, only the number of segments not yet translated is shown. This column should show “0” once the translation is finished.

Changing the layout

This section describes what you can change with regard to the layout of a report.

To begin, click the “Settings” tab in the “Create Counting Report” window to display the “Settings” tabbed page (see [Figure 141](#)).

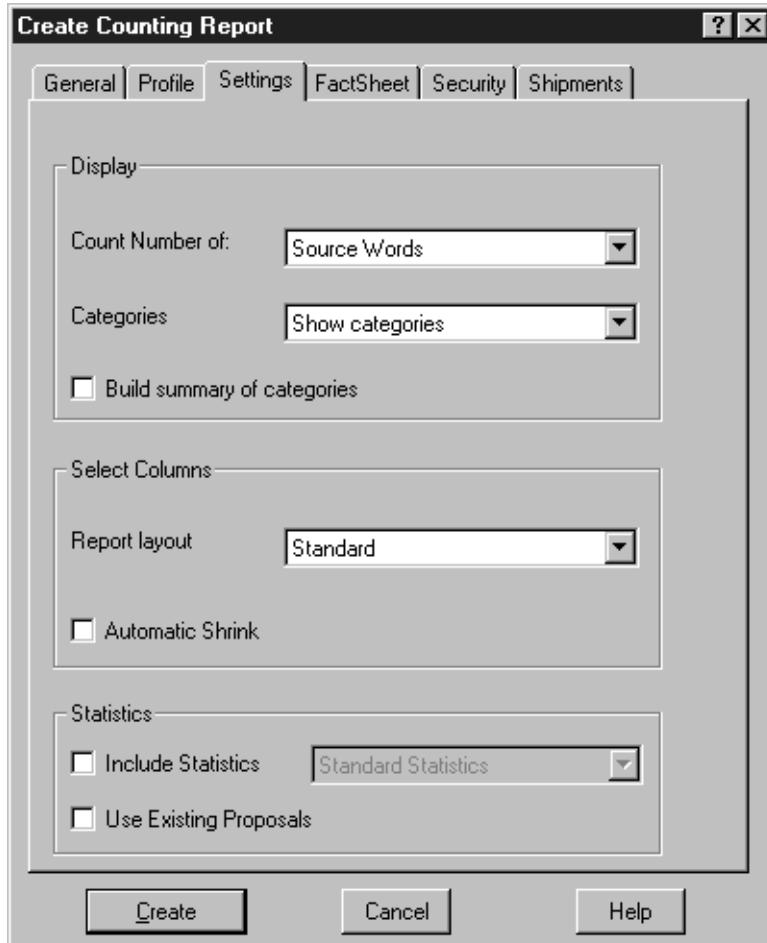


Figure 141. Create Counting Report window — Settings page

On this page, you can choose or specify the options and parameters described in the following. A Calculation Report (Base) is used as an example to demonstrate the effect that the different specifications have on the report layout.

Count number of

Specify what is to be counted to calculate the effort and payment of the translator:

Source words

The source words are used for the calculation. This is the preferred setting.

Differences in the complexity of different languages can be taken into account using language-dependent complexity factors on the "FactSheet" page of the "Create Counting Report" window (see [Setting the factors for cost calculation](#)).

Target words

The target words are used for the calculation. Select this option only for tracking and informational purposes.

Segments

The translated segments are used for the calculation.

Modified words

The modified source words are used for the calculation.

Categories

Specify whether your report is to contain a "Category" column breaking your calculation down to segments with 1..4 words, 5..14 words, and more than 15 words.

Show categories

The report contains a "Category" column. The following Calculating Report contains such a column:

Document		DEVICE.SCR															
Folder		C:\EQFSAMPLE1.FOO															
		Source Words															
		Number			Analyze			Edit			Fuzzy						
Doc Id	Document	Shipm	Catego.		Auto	Post	Edit	Exact	Replace		50-70%	70-90%	>90%	Mach.	None	Not	
1	DEVICE.SCR	1	1..4			3										4	
			5..14			44										5	31
			>= 15			84											

Figure 142. Calculating Report (Base), with categories

Hide categories

The report does not contain a "Category" column but shows the calculation summed up in one row. The following Calculating Report differs from [Figure 142](#) in that it does not contain any categories:

Document		DEVICE.SCR															
Folder		C:\EQFSAMPLE1.FOO															
		Source Words															
		Number			Analyze			Edit			Fuzzy			Mach.	None	Not	
Doc Id	Document	Shipm	Catego.		Auto	Post	Edit	Exact	Replace		50-70%	70-90%	>90%	Matches	Matches	Transl	
1	DEVICE.SCR	1		131												5	35

Figure 143. Calculating Report (Base), without categories

Build summary of categories

If you selected "Show categories", you can select this option to add a row containing the sum of the calculation. It is similar to the row you get when specifying "Hide categories". Your Calculating Report would then look as follows:

Document		DEVICE.SCR															
Folder		C:\EQFSAMPLE1.FOO															
		Source Words															
		Number			Analyze			Edit			Fuzzy			Mach.	None	Not	
Doc Id	Document	Shipm	Catego.		Auto	Post	Edit	Exact	Replace		50-70%	70-90%	>90%	Matches	Matches	Transl	
1	DEVICE.SCR	1	1..4			3										4	
			5..14			44										5	31
			>= 15			84											
	DEVICE.SCR	1	Sum		131											5	35

Figure 144. Calculating Report (Base), with summary of categories

Report layout

Specify which columns the report is to contain:

Standard

The report contains the following columns:

- Analyze Auto
- Analyze Post
- Analyze Edit
- Edit Exact
- Edit Replace
- Fuzzy 50–70%
- Fuzzy 70–90%
- Fuzzy >90%
- Machine matches
- Manually translated (None Matches)
- Not Translated

These columns are described in [The report layout](#).

Standard and group summary

The report contains the columns previously described and summaries for the Analyze, Edit, and Fuzzy columns. Your Calculating Report would then look as follows:

Document		DEVICE.SCR												
Folder	C:\EQF\SAMPLE1.FOO	Source Words												
Doc Id	Document	Shipm	Catego.	Sum	Analyze		Sum	Edit		Sum	Fuzzy			
					Analyze	Auto	Post	Edit	Exact	Replace	Fuzzy	50-70%	70-90%	>90%
1	DEVICE.SCR	1	1..4	3		3								
			5..14	44		44								
			>= 15	84		84								
	DEVICE.SCR	1	Sum	131		131								

Figure 145. Calculating Report (Base), standard and group-summary layout

Shrunk to groups

For the Analyze, Edit, and Fuzzy columns, the report only shows the summaries to improve readability. Your Calculating Report would then look as follows:

Document		DEVICE.SCR								
Folder	C:\EQF\SAMPLE1.FOO	Source Words								
Doc Id	Document	Shipm	Catego.	Sum	Analyze	Edit	Fuzzy	Mach.	None	Not
					Analyze	Post	Exact	Matches	None	Transl
1	DEVICE.SCR	1	1..4	3						4
			5..14	44					5	31
			>= 15	84						
	DEVICE.SCR	1	Sum	131					5	35

Figure 146. Calculating Report (Base), shrunk to groups

Automatic shrink

Select this option if you want your report without the columns containing no values. [Figure 6](#) would then look as follows:

Document		DEVICE.SCR						
Folder	C:\EQF\SAMPLE1.FOO	Source Words						
Doc Id	Document	Shipm	Catego.	Sum	Analyze	Post	None	Not
					Analyze	Post	None	Not
1	DEVICE.SCR	1	1..4	3		3		4
			5..14	44		44	5	31
			>= 15	84		84		
	DEVICE.SCR	1	Sum	131		131	5	35

Figure 147. Calculating Report (Base), standard and group-summary layout, automatic shrink

Include statistics

This option is only available to the Calculating Report.

Select this option if you want to add a statistics table of your project to the end of your report. This table can help you check the quality and consistency of the translation. A low percentage of matches used indicates that the translator used few of the proposals offered by **OpenTM2**.

You can choose between the following types of statistics:

Standard statistics

The statistics table contains the percentage of proposals offered (exact, replace, fuzzy, and machine proposals) that were actually used by the translator. Such a statistics table can look as follows:

Statistics										
Document Folder		TRANS.DOC C:\EQF\SAMPLE1.FOO								
Doc Id	Document	Number	Edit			Sum	Fuzzy			Mach
		Shipm	Catego.	Exact	Repl	Fuzzy	50-70%	70-90%	>90%	Matches
1	TRANS.DOC	1	1..4	-	-	-	-	-	-	-
			5..14	62%	-	100%	-	100%	-	-
			>= 15	100%	-	-	-	-	-	-

Figure 148. Standard statistics

Advanced statistics

The statistics table contains both the percentage of proposals used and the absolute number of proposals not used during the translation. This option enables you to estimate whether the number of matches not copied by the translator is acceptable. Your table can look as follows:

Statistics															
Document Folder		TRANS.DOC C:\EQF\SAMPLE1.FOO													
Doc Id	Document	Number	Edit			Sum	Mach.	Edit			Sum	Fuzzy			Mach
		Shipm	Catego.	Exact	Replace	Fuzzy	Matches	Exact	Repl	Fuzzy	50-70%	70-90%	>90%	Matches	
1	TRANS.DOC	1	1..4					-	-	-	-	-	-	-	
			5..14	18		9		62%	-	100%	-	100%	-	-	
			>= 15	22				100%	-	-	-	-	-	-	

Figure 149. Advanced statistics

Use existing proposals

Select this option if you want your report to show all proposals offered by **OpenTM2** regardless of whether the translator accepted and copied them. This option enables you to calculate the effort based on the assumption that all proposals were accepted.

Setting the factors for cost calculation

Documents can differ considerably with regard to their complexity. Some documents are easy to translate while the translation of others is difficult and time-consuming. The complexity of a document can also vary depending on the source language. **OpenTM2** enables you to take into account such language-dependent differences.

For each translation project you can specify different complexity and pay factors. The factors can be reflected in the History, Calculating, Preanalysis, and Redundancy Reports.

Click the "FactSheet" tab of the "Create Counting Report" window to display the "FactSheet" tabbed page (see [Figure 150](#)).

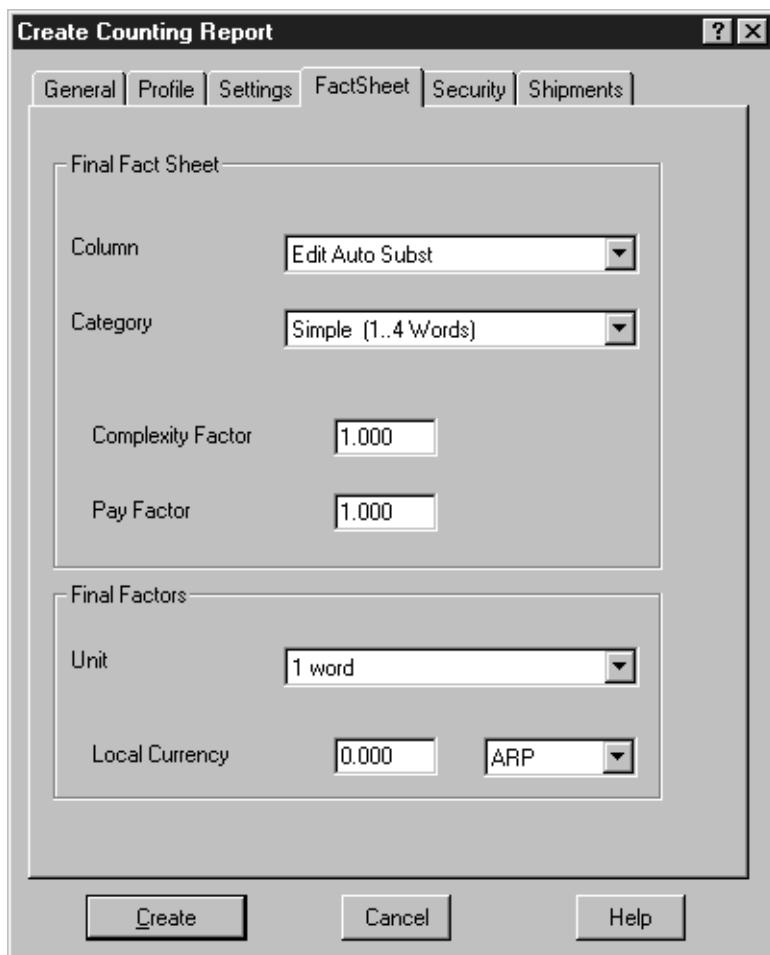


Figure 150. Create Counting Report window — FactSheet page

On this page, you can choose or specify the following options and parameters:

Column

Select the calculating factor for which you want to choose different complexity and pay factors. You can choose one of the following:

Analysis Auto Subst

Contains the number of exact and exact-exact matches retrieved from a **Translation Memory** during the analysis of the document.

Analysis Auto Subst 2

Has the same contents as, and replaces, the “Analysis Auto Subst” column after the analyzed document is opened for translation.

Edit Auto Subst

Contains the number of exact matches retrieved from a **Translation Memory** during the translation of the document using the “Automatic substitution” option from the **Translate** menu.

Exact Matches

Contains the number of exact matches copied from a **Translation Memory** during the translation of the document using the **Ctrl+n** key.

Replace Matches

Contains the number of fuzzy replacement matches copied from a **Translation Memory** during the translation of the document.

Fuzzy Matches

Contains the number of fuzzy matches copied from a **Translation Memory** during the translation of the document. The fuzzy matches are broken down to matches that fit less than 70%, less than 90%, and over 90%.

Machine Matches

Contains the number of exact matches copied from an **Initial Translation Memory** during the translation of the document.

Manually Translated

Contains the number of segments to be manually translated.

For a more detailed description of these options, refer to [The report layout](#). You can change the complexity and pay factors for several or all columns of a report. For example, if you want to specify different complexity and pay factors for the Exact Matches and Manually Translated columns, proceed as follows:

1. Select Exact Matches.
2. Select a category.
3. Specify a complexity factor.
4. Specify a pay factor.
5. Select Manually Translated. At this point of time, **OpenTM2** saves your settings for the Exact Matches column.
6. Select a category and specify a complexity factor and a pay factor.

Categories

Select the segments for which you want to choose different complexity and pay factors. You can choose one of the following:

Simple (1..4 Words)

Segments with one to four words.

Medium (5..14 Words, mean)

Segments with five to 14 words.

Complex (>= 15 Words)

Segments with more than 15 words.

Complexity factor

Specify a complexity for the selected column and category. This option enables you to take into account the translation challenges that each language poses.

For example, a Finnish sentence with more than 15 words might be more difficult to translate than an English sentence of the same length. So, for a translation from Finnish, you might want to specify a complexity factor of 2 for segments with more than 15 words that need to be manually translated, whereas for a translation from English, you might want to leave the complexity factor at 1.

The default complexity factor is 1.

Pay factor

Specify a number by which the amount of money defined in the **Local Currency** fields is to be multiplied for the unit specified.

It is recommended that you specify the same pay factor for an entire translation project.

The default pay factor is 1.

Unit

Select the unit on which your payment is based:

1 word

Payment is per word.

Standard line (10 words)

Payment is per line, which consists of 10 words on average.

Standard page (250 words)

Payment is per page, which consists of 250 words on average.

Local currency

Specify the amount of money to be paid for each unit and the currency.

Example:

The following figure shows the fact sheet of a Calculation Report that was created after the translation was finished:

Document Folder		Final Fact Sheet C:\EQF\SAMPLE1.FOO Source Words					
Doc Id	Document	Number	Shipm	Analyze	Edit	Fuzzy	None
		Category	Post	Exact	70-90%	Matches	
Actual Words	1..4	3					4
	5..14	44	5	5	5		26
	>= 15	84					
Actual Words	Sum	131	5	5	5	30	
Complexity	1..4	0.50	0.75	1	1		
Factor	5..14	0.50	0.75	1	1	2	
	>= 15	0.50	0.75	1	1	3	
mean comp	Sum	0.50	0.75	1	1	2	
Complexity	1..4	1				4	
Factor *	5..14	22	3	5	5	52	
Actual Words	>= 15	42					
Compl. Word	Sum	65	3	5	5	56	
Pay	1..4	1	1	1	1	1	
Factor	5..14	1	1	1	1	1	
	>= 15	1	1	1	1	1	
mean pay	Sum	1	1	1	1	1	
Pay	1..4	1				4	
Factor *	5..14	22	3	5	5	52	
Compl. Word	>= 15	42					
Payable Words	Sum	65	3	5	5	56	

Final Summary	
Payable Words	130.25
Standard Lines	13.02
Local Currency	10.00 USD
Total Pay	130.25 USD

Figure 151. Calculating Report — Fact Sheet

To achieve this result, you specify the following on the "FactSheet" page:

1. Select **Analysis Auto Subst2** as column.
2. Select **Simple (1..4 Words)** as category.
3. Specify **0,5** for the complexity.
4. Select **Medium (5..14 Words, mean)** as category.
5. Specify **0,5** or the complexity.
6. Select **Complex (>= 15 Words)** as category.
7. Specify **0,5** for the complexity.
8. Select **Exact Matches** as column.
9. Select **Simple (1..4 Words)** as category.
10. Specify **0,75** for the complexity.
11. Select **Medium (5..14 Words, mean)** as category.
12. Specify **0,75** for the complexity.
13. Select **Complex (>= 15 Words)** as category.
14. Specify **0,75** for the complexity.
15. Select **Manually Translated** as column.
16. Select **Simple (1..4 Words)** as category.
17. Specify **1** for the complexity.
18. Select **Medium (5..14 Words, mean)** as category.
19. Specify **2** for the complexity.
20. Select **Complex (>= 15 Words)** as category.
21. Specify **3** for the complexity.
22. Specify Standard line (10 words) as unit.
23. Specify **10 USD** as local currency.

OpenTM2 calculates the costs by multiplying each category with the complexity factor specified for that category. Each result is multiplied with the pay factor specified. The sum is the number of payable words. Because the unit is defined to be a standard line, the number of payable words is divided by 10. The result is multiplied by the cost ("Local currency") specified for a standard line. You then receive the costs for the translation of this document.

Working from the command area (EQFCMD)

Using the OpenTM2 windows is only one way of performing translation tasks with this product. An alternative way is to type instructions directly in the system's command area.

To start a task from the command area:

1. Type a command and its parameters in your operating system's command area (next to the command prompt) according to the command syntax described in the following chapters.
2. Press Enter.

Note that lowercase letters in a parameter name are optional and need not be typed. A mixture between uppercase and lowercase letters is allowed. However, a muddled letter sequence like iNOverWRiT3 must not be used. The parameters can be entered in any order whatever.

To process several commands in sequence, type the individual commands into a batch file (*.BAT or *.CMD) using any text editor. You can execute these commands by starting the batch file.

To avoid retyping of long text strings in commands, you can also type a command and its parameters in a text editor, which is capable of the **Copy to clipboard** function. You can then **paste** the text string from the clipboard directly into your system's command area and start the task.

How to read syntax diagrams

In this chapter diagrams are used to illustrate the programming syntax. To use a diagram, follow a path from left to right, top to bottom, adding elements as you go. In these diagrams, all spaces and other characters are significant.

Each diagram begins with a double right arrowhead and ends with a right and left arrowhead pair. Lines beginning with single right arrowheads are continuation lines.

keyword=variable_value

Keywords are all in lowercase, but can be entered in uppercase or in lowercase. Variable values that you provide are shown in *italics* and are usually in lowercase. Where values are shown in uppercase, they should be entered as they appear.

In a choice of items, the default item is always shown above the main line:

keyword= default_value other_value other_value

Optional syntax elements are shown below the main line:

keyword=value

A repeat arrow shown above an item or a stack of items indicates that you can specify the item multiple times or specify more than one of the items. A character (such as a comma) on the repeat arrow line indicates that the items must be separated by that character. A repeatable operand is shown like this:

keyword= (,variable_value)

Syntax diagrams can be broken into fragments. A fragment is indicated by vertical bars with the name of the fragment between the bars. The fragment is shown following the main diagram, like so:

/KEYWORD= ITEM1 variable1 variable2 variable3

Analyzing a document or folder

Purpose

documentanalyzing (from command area) analyzingdocuments (from command area) To analyze a document or a set of documents it might be more convenient to use a command instead of OpenTM2 windows. It is particularly useful if you have many large documents or folders that you want to analyze overnight.

Format

EQFCMDanalyzing documents commandsEQFCMDanalyzing a document or folder The following syntax diagram describes how you analyze documents from the Windows (R) command line. Note that lowercase letters in a parameter name are optional and need not be typed.

**EQFCMD /TAsk=ANALYSIS /FLD=folder /FIlles= document_name (,document_name)
@list_file_name /OPtions= (,) /MEm= memdb¹ (,memdb) /OVerwrite= NO YES /QUIET option
ADDTOMEM AUTO AUTOCONTEXT AUTOLAST AUTOJOIN TMMATCH UNTRANSLATED
ADJUSTLEADWS ADJUSTTRAILWS PROTECTXMPSCREEN RESPECTCRLF**

1 Mandatory only when Options=ADDTOMEM

Parameters

/TAsk=ANALYSIS

ANALYSIS specifies that you want to analyze documents.

/FLD=folder

folder specifies the name of a folder that contains the documents that you want to analyze.

/FIlEs=document_name

document_name specifies the name of a document to be analyzed.

If you specify several documents, enclose the document names in brackets and separate them by commas.

If you want to specify a list of document names in a list file, refer to this list file by preceding the list file name with the @ symbol (@ list_file_name). In the list file enclose the document names in brackets and separate them by commas.

If you do not specify the **FIlEs** option, all documents in the folder are analyzed.

/OPtions=

This parameter is particularly useful if you have Translation Memory databases available from previous translations. It specifies the following:

ADDTOMEM

OpenTM2 can search existing Translation Memory databases for segments that match segments in the document being analyzed. Segments that match exactly or almost exactly are copied into the Translation Memory associated to the document or folder being analyzed. In that way, you can start to fill a document-specific Translation Memory even before its translation has begun.

AUTO

You can choose to have Translation Manager automatically replace sentences with matches that already exist in the Translation Memory. This substitution is done only for exact matches. The document is then already partially translated when you start working with it, enabling you to focus on the translation of new segments.

AUTOCONTEXT

See the explanation for **AUTO**. However, if more than one exact match has been found, the translation of the segment is to be taken from the same document.

AUTOLAST

See the explanation for **AUTO**. However, if more than one exact match has been found, the most recent translation of the segment is to be taken.

AUTOJOIN

Joins neighboring segments. This can be useful if you want to analyze a document that you have worked with before and where you joined neighboring text segments before you translated them. During a new analysis run these segments are treated as separate units again. However, in the associated Translation Memory there is a match (translation) for the joined segments only. To find this match more easily during automatic substitution, this option joins neighboring segments in the document to be analyzed. Note that neighboring segments are only joined if for the first segment a fuzzy match has been found.

TMMATCH

OpenTM2 counts the number and type of matches found in the **Translation Memory** for the document. The document is then segmented and the matches are counted, but no segments are replaced with matches found in the **Translation Memory**.

If you also specify **AUTO**, **OpenTM2** automatically replaces segments with matches found in the **Translation Memory** and counts the number and type of matches replaced.

UNTRANSLATED

All untranslated segments are to be stored in a separate file.

ADJUSTLEADWS

During whitespace handling of automatic substitution, leading whitespaces are adjusted to whitespaces in source segment.

ADJUSTTRAILWS

During whitespace handling of automatic substitution, trailing whitespaces are adjusted to whitespaces in source segment.

RESPECTCRLF

During whitespace handling of automatic substitution, substitution takes place with respect of CRLF in the segment.

PROTECTXMPSCREEN

When specified the <xmp> and <screen> sections in DITA and IDDOC are protected and excluded from translation

/MEm=memdb

memdb specifies a Translation Memory or a list of Translation Memory databases that is to be searched when **ADDTOMEM** has been specified.

/OVerwrite=

Specifies if an existing document will be overwritten when its translation has already been started. (**NO** is the default).

/QUIET

If you specify this parameter, you are not prompted with any message window. If you do not specify this parameter, a message window pops up whenever an error occurs or when the task has successfully completed.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to analyze a document by using the EQFCMD command.

Example

```
eqfcmd /task=analysis /fld=myfol /files=16.scr  
/options=(addtomem,auto,untranslated) /mem=oldmem
```

In this example, the document to be analyzed has the file name **16.scr** and is stored in folder **myfol**. Any matches found in the Translation Memory **oldmem** are to be added to the Translation Memory of folder **myfol**. Any original segment for which an exact match has been found is to be replaced with its corresponding translation segment. All the remaining segments that are not translated are to be stored in a separate file.

Exporting documents

Purpose

documentexporting (from command area) exportingdocuments (from command area) If you have many large documents that you want to export, using the command area of your operating system instead of OpenTM2 windows might be more convenient.

Before exporting, ensure that:

- **OpenTM2** has been started.
- A folder exists that contains the documents to be exported.

The properties of the folder or of the documents give information on the used markup table and other document-related information that is necessary when exporting the respective documents.

To work from the command area, use the **EQFCMD** command. Its syntax is described in the following chapter.

Format

EQFCMD exporting documents commands The following syntax diagrams describe how you export a document from the Windows^(R) command area. Note that lowercase letters in a parameter name are optional and need not be typed.

```
EQFCMD /TAsk=DOCEXP /FLD=folder /SStartpath=startpath /FFiles= file_name (,file_name )  
@list_file_name /OPtions= TARGET SOURCE SNOMATCH /OVerwrite= NO YES /QUIET
```

Parameters

/TAsk=DOCEXP

DOCEXP specifies that you want to export a document.

/FLD=*folder*

folder specifies the name of the folder containing the documents that you want to export.

/SStartpath=*startpath*

startpath specifies the drive, directory, or both where the file exported is to be placed. The path information specified becomes part of, that is it prefixes, the name of the document. The directory specified must already exist.

If you omit this parameter, the document is placed in the drive and directory specified in *file_name*. However, this drive and directory does not become part of the document name.

/FFiles=*file_name*

file_name specifies the name of the document you want to export. The *file_name* specification can include the drive and directory where the file exported is to be placed. The subdirectory specified here must already exist.

You can use the wildcard characters '*' and '?' in the name part of the parameter to export all documents matching the given name pattern (e.g. "C:\DOCEXP*.DOC" to export all documents with a file name extension of ".DOC" to the directory "C:\DOCEXP").

If you specify several documents, enclose the document names in brackets and separate them by commas.

If you want to specify a list of document names in a list file, refer to this list file by preceding the list file name with the @ symbol (@ *list_file_name*). In the list file enclose the document names in brackets and separate them by commas.

/OPtions=

Specifies whether you want to export an original document (SOURCE), a translation (TARGET), or a document with nonmatching segments.

/OVerwrite=

Specifies whether an existing document is overwritten when exporting a document with the same name (NO is the default).

/QUIET

If you specify this parameter, you are not prompted with any message window. If this option is omitted, a message window pops up whenever an error occurs or when the task has successfully completed.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters.

Examples

The following example shows how to export a document using the EQFCMD command with the TASK option.

Example

```
eqfcmd /task=docexp /fld=pharma /files=c:\med.txt /options=target  
/overwrite=yes
```

In this example, the document to be exported is called `med.txt`. It is contained in folder `pharma`. The translation of `med.txt` is exported to the root directory of drive C. If `med.txt` already exists, it is overwritten.

The following example shows how to export documents in external format with a path.

Example

```
eqfcmd /task=docexp /fld=sample1 /startpath=i:\transl  
/files=(test1\demo1.scr,test1\demo2.scr)
```

The document `test1\demo1.scr` is exported to `i:\transl\test1\demo1.scr` and the document `test1\demo2.scr` is exported to `i:\transl\test1\demo2.scr`.

Importing documents

Purpose

If you have many large documents that you want to import, using the command area of your operating system instead of OpenTM2 windows might be more convenient.

Before importing, ensure that:

- **OpenTM2** has been started.
- A folder has been created. This folder can still be empty and will later contain the documents to be imported.

The properties of the folder give information on the used markup language and other document-related information that is necessary when importing the respective documents.

To work from the command area, use the **EQFCMD** command. Its syntax is described in the following chapter.

Format

The following syntax diagram describes how to import from the command area of your operating system. Note that lowercase letters in a parameter name are optional and need not be typed.

EQFCMD /TAsk=DOCIMP /FLD=folder /STartpath=startpath /FFiles= file_name (.file_name) @list_file_name /ALIAS=alias /OVerwrite= NO YES /QUIET /EDit=editor /MArket=markup /CONV=conv

Parameters

/TAsk=DOCIMP

DOCIMP specifies that you want to import a document.

/FLD=folder

folder specifies the name of the folder where the documents imported are to be placed.

/STartpath=startpath

startpath specifies the drive, directory, or both, to be omitted from the document's name. It only applies to documents imported in external format where the path in which the document resides normally becomes part of the document's name.

/FFiles=file_name

file_name specifies the name of the document you want to import. The file_name specification can include the drive and directory where the file to be imported is found.

If you specify several documents, enclose the document names in brackets and separate them by commas.

If you want to specify a list of document names in a list file, refer to this list file by preceding the list file name with the @ symbol (@ list_file_name). In the list file enclose the document names in brackets and separate them by commas.

/ALIAS=alias

alias specifies an alias name for the document to be imported. If you import several documents, this alias name applies to all of them. The name can be up to 256 characters long.

/OVerwrite=

Specifies if an existing document will be overwritten when importing a document with the same name (NO is the default).

/QUIET

If you specify this parameter, you are not prompted with any message window. If this option is omitted, a message window pops up whenever an error occurs or when the task has successfully completed.

/EDit=editor

Specifies the editor to be used for the imported document.

/MArket=markup

Specifies the markup table to be used for the document.

/CONV=conv

Specifies the conversion to be used for the document.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to import several documents using the EQFCMD command.

Example

```
eqfcmd /task=docimp /fld=pharma /files=@medlist.lst /overwrite=no
```

In this example, the documents to be imported are listed in file `medlist.lst`. `medlist.lst` looks as follows:

`medlist.lst`

```
(d:\medset.doc, d:\medfrm.doc, d:\medb01.doc, d:\medb02.doc)
```

The file `medlist.lst` must be located either on the current directory, or (if it is not in the current directory) you must specify the complete path where it is located. For example,
`@d:\test\medlist.lst`.

Note:

There is a @ sign before the drive letter.

All files contained in `medlist.lst` are to be imported into folder `pharma` and are located in the root directory of drive D. If one of the files already exists in the specified folder, it is not overwritten and you get a message.

Examples

The following examples show how to import documents in external format with a path. Assume that the following directories and files are on your V disk:

```
[proj1]
[vers1]
[mri]
base.mri
dialog.mri
dialog2.mri
message.osm
[docs]
intro.htm
lesson.htm
[vers2]
[mri]
base.mri
dialog.mri
dialog2.mri
message.osm
[docs]
intro.htm
lesson.htm
```

Example 1

```
eqfcmd /task=docimp /fld=test /startpath=v:\  
/files=(vers1\docs\*.htm,vers2\docs\*.htm)
```

In this example, the following documents are imported into folder `test`:

```
vers1\docs\intro.htm
vers1\docs\lesson.htm
vers2\docs\intro.htm
vers2\docs\lesson.htm
```

Example 2

```
eqfcmd /task=docimp /fld=test /startpath=v:\vers1\ /files=(mri\*,*,docs\*)
```

In this example, the following documents are imported into folder `test`:

```
mri\base.mri  
mri\dialog.mri  
mri\dialog2.mri  
mri\message.osm  
docs\intro.htm  
docs\lesson.htm
```

Example 3

```
eqfcmd /task=docimp /fld=test /startpath=v:\vers1\ /files=(.*)
```

In this example, the following documents are imported into folder test:

```
mri\base.mri  
mri\dialog.mri  
mri\dialog2.mri  
mri\message.osm  
docs\intro.htm  
docs\lesson.htm
```

Example 4

```
eqfcmd /task=docimp /fld=test /startpath=v:\ /files=(.*)
```

In this example, the following documents are imported into folder test:

```
vers1\mri\base.mri  
vers1\mri\dialog.mri  
vers1\mri\dialog2.mri  
vers1\mri\message.osm  
vers1\docs\intro.htm  
vers1\docs\lesson.htm  
vers2\mri\base.mri  
vers2\mri\dialog.mri  
vers2\mri\dialog2.mri  
vers2\mri\message.osm  
vers2\docs\intro.htm  
vers2\docs\lesson.htm
```

Example 5

```
eqfcmd /task=docimp /fld=test /startpath=v:\vers1\docs /files=(*.htm)
```

In this example, the following documents are imported into folder test:

```
intro.htm  
lesson.htm
```

Deleting documents

Purpose

If you want to delete a document, using a command typed in your system's command area instead of OpenTM2 windows might be quicker and more convenient.

Format

The following syntax diagram describes how you delete documents with the EQFCMD command. Note that lowercase letters in a parameter name are optional and need not be typed.

EQFCMD /TAsk=DOCDEL /FLD=*folder* /FIles= *file_name* (*file_name*) @*list_file_name* /QUIET

Parameters

/TAsk=DOCDEL

DOCDEL specifies that you want to remove documents.

/FLD=*folder*

Specifies the name of the folder that contains the documents to be removed.

/FIles=*file_name*

file_name specifies the name of the document you want to remove. The *file_name* specification

can include the drive and directory where the file to be removed is found.

If you specify several documents, enclose the document names in brackets and separate them by commas.

If you want to specify a list of document names in a list file, refer to this list file by preceding the list file name with the @ symbol (@ *list_file_name*). In the list file enclose the document names in brackets and separate them by commas.

/QUIET

If you specify this parameter, you are not prompted with any message window. If you do not specify this parameter, a message window pops up whenever an error occurs or when the task has successfully completed.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to delete documents using the EQFCMD command.

Example

```
eqfcmd /task=docdel /fld=pharma /files=(med.txt) /quiet
```

In this example, the document `med.txt` is removed from folder `pharma` without prompting the user.

Creating a folder

Purpose

foldercreating (from command area) creatingfolders (from command area) If you need to set up new folders frequently, using the command area of your operating system instead of OpenTM2 windows might be more convenient.

To work from the command area, use the **EQFCMD** command. Its syntax is described in the following chapter.

Format

EQFCMDcreating folders commands**EQFCMD**creating folders The following syntax diagram describes how you create a folder from the Windows^(R) command area. Note that lowercase letters in a parameter name are optional and need not be typed.

```
EQFCMD /TAsk=FLDCRT
/NAme=folder
/DEsc=desc /TOdrive=drive /MEm=memdb /MArkup=markuptable /EDit=editor /Dlct=dict (dict)
/SRcIng=source /TGtIng=target /QUIET
```

Parameters

/TAsk=FLDCRT

FLDCRT specifies that you want to set up a new folder.

/NAme=*folder*

folder specifies the name of the new folder.

/DEsc=*desc*

desc specifies the description of the new folder (up to 40 characters). If none is specified, the folder will have no description and will be more difficult to relate to a specific project later on.

/TOdrive=*drive*

drive specifies the drive where the new folder is to be placed. It must be either the main drive or a drive that has been configured as additional drive.

/MEm=*memdb*

memdb specifies the name of an existing Translation Memory that is to be related to the new folder.

/MArkup=*markuptable*

markuptable specifies the name of a markup table that is to be related to the markup language of

the documents that will be stored in the new folder.

/EDit=editor

editor specifies the name of an editor that is to be associated with the new folder (only STANDARD is possible).

/DIct=dict

dict specifies the name of a dictionary to be associated with the new folder. This parameter is optional, that is, it is not required that a dictionary is specified. If you specify several dictionaries, enclose the dictionary names in brackets and separate them by commas.

/SRcIng=source

source is the source language of the documents stored in the new folder.

/TGtIng=target

target is the target language of the documents stored in the new folder.

/QUIET

If you specify this parameter, you are not prompted with any message window. If you do not specify this parameter, a message window pops up whenever an error occurs or when the task has successfully completed.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to create a folder using the EQFCMD command.

Example

```
eqfcmd /task=fldcrt /name=proj1 /desc="new folder" /todrive=e  
/mem=ibmmem /markup=eqfbook /edit=standard /dict=(ibmterm,ibm2)  
/srclng=German(national) /tgtlng=English(U.S.)
```

In this example, the folder to be created is called `proj1` and is described as `new folder`. It will be located on drive E and will have the following translation resources attached to it:

- The Translation Memory `ibmmem`
- The dictionaries `ibmterm` and `ibm2`
- The markup table `eqfbook`
- The editor `standard`

The source language of the folder's documents will be German (national), the target language will be English (U.S.).

Exporting a folder

Purpose

folderexporting (from command area) exportingfolders (from command area) If you need to set up new folders frequently, using the command area of your operating system instead of OpenTM2 windows might be more convenient.

To work from the command area, use the **EQFCMD** command. Its syntax is described in the following chapter.

Format

EQFCMD **Exporting folders commands** The following syntax diagram describes how you export a folder from the Windows^(R) command area. Note that lowercase letters in a parameter name are optional and need not be typed.

```
EQFCMD  
/Task=FLDEXP  
/FLD=folder /FRomdrive=drive /FromPath=\EQF\EXPORT /FromPath=path /FIles=  
document_name (,document_name )  
@list_file_name /TOdrive=drive /ToPath=\EQF\EXPORT /ToPath=path /OOptions= (,  
) /OVerwrite= YES NO /DEsc= desc @desc_file_name /QUIET option DICT MEM ROMEM  
DOCMEM DELETE
```

Parameters

/TAsk=FLDEXP

FLDEXP specifies that you want to export a folder.

/FLD=folder

folder specifies the name of the folder that you want to export.

/FRomdrive=drive

drive specifies the drive where the folder is to be exported from. The default is the drive where OpenTM2 is installed.

/FromPath=path

path specifies the path where the folder is to be exported from. Start the path value with a backslash \. The default is \eqf\export.

/FFiles=document_name

document_name specifies the name of a document to be exported.

If you specify several documents, enclose the document names in brackets and separate them by commas.

If you want to specify a list of document names in a list file, refer to this list file by preceding the list file name with the @ symbol (@ *list_file_name*). In the list file enclose the document names in brackets and separate them by commas.

If you do not specify the **FFiles** option, all documents in the folder are exported.

/TOdrive=drive

drive specifies the drive where the folder is to be exported to.

/ToPath=path

path specifies the path where the folder is to be exported to. Start the path value with a backslash \. The default is \eqf\export.

/OOptions=

Specifies whether you want to export the folder with its dictionaries (DICT), with its Translation Memory (MEM), with its Translation Memory databases to be searched (ROMEM), with its **Translation Memory** and all the **Translation Memory databases** defined for its documents (DOCMEM) , or whether you want to delete the folder after it has been exported (DELETE). If you specify more than one option, you must enclose them in brackets.

/OOverwrite=

Specifies whether an existing folder is overwritten when exporting a folder with the same name (NO is the default).

/DEsc=desc

desc specifies the description that you want to add to the exported folder, or the name of a file containing the description. The text that gives more information about the folder must be enclosed in quotes. If you specify a file that contains the text, the name of the file must be preceded by the @ symbol.

/QUIET

If you specify this parameter, you are not prompted with any message window. If you do not specify this parameter, a message window pops up whenever an error occurs or when the task has successfully completed.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to export a folder using the EQFCMD command.

Example

```
eqfcmd /task=fldexp /fld=pharma /files=(161.scr,1500.scr) /todrive=e  
/options=(dict,mem) /desc="This is information on the folder"
```

In this example, the folder to be exported is called *pharma*. It is exported to drive E with the

documents 161.scr and 1500.scr, with its associated dictionary and Translation Memory. The note This is information on the folder is added to the folder.

Importing a folder

Purpose

It might be more convenient to use a command instead of OpenTM2 windows to import a folder. This is especially useful if you have many folders that you want to import.

Format

The following syntax diagram describes how you import a folder from the Windows^(R) command line. Note that lowercase letters in a parameter name are optional and need not be typed.

```
EQFCMD  
/TAsk=FLDIMP  
/FLD=folder /FRomdrive=drive /FromPath=\EQF\EXPORT /FromPath=path /Options=(  
)  
/TOdrive=drive  
/ToPath=\EQF\EXPORT /ToPath=path /QUIET /EDit=editor /MArket=markup /CONV=conv  
option DICT MEM
```

Parameters

/TAsk=FLDIMP

FLDIMP specifies that you want to import a folder.

/FLD=*folder*

folder specifies the name of the folder that you want to import.

/FRomdrive=*drive*

drive specifies the drive where the folder is imported from.

/FromPath=*path*

path specifies the path where the folder is imported from. Start the path value with a backslash \. The default is \eqf\export.

/Options=

Specifies whether you want to import the folder with its associated dictionary (DICT) or Translation Memory (MEM). If you specify more than one option, you must enclose them in brackets and separate them with commas.

/TOdrive=*drive*

drive specifies the drive where the folder is to be imported to. If nothing is specified, the folder is imported to the drive where the OpenTM2 program files are located.

/ToPath=*path*

path specifies the path where the folder is to be imported to. Start the path value with a backslash \. The default is \eqf\export.

/QUIET

If you specify this parameter, you are not prompted with any message window. If you do not specify this parameter, a message window pops up whenever an error occurs or when the task has successfully completed.

/EDit=*editor*

Specifies the editor to be used for the imported folder.

/MArket=*markup*

Specifies the markup table to be used for the imported folder.

/CONV=*conv*

Specifies the conversion to be used for the imported folder.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to import a folder using the EQFCMD command.

Example

```
eqfcmd /task=fldimp /fld=pharma /fromdrive=a /todrive=e /options=(dict,mem)
```

In this example, the folder to be imported is called `pharma`. It is imported from drive A to drive E together with its associated dictionary and Translation Memory.

Deleting a folder

Purpose

folderdeleting (from command area) deletingfolders (from command area) If you want to delete a folder, using a command typed in your system's command area instead of OpenTM2 windows might be quicker and more convenient.

Format

EQFCMDdeleting folders commandsEQFCMDdeleting folders The following syntax diagram describes how you delete a folder with the EQFCMD command. Note that lowercase letters in a parameter name are optional and need not be typed.

EQFCMD /TAsk=FLDDEL /FLD=*folder* /QUIET

Parameters

/TAsk=FLDDEL

`FLDDEL` specifies that you want to remove a folder.

/FLD=*folder*

Specifies the name of the folder that you want to remove.

/QUIET

If you specify this parameter, you are not prompted with any message window. If you do not specify this parameter, a message window pops up whenever an error occurs or when the task has successfully completed.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to delete a folder using the EQFCMD command.

Example

```
eqfcmd /task=foldel /fld=pharma /quiet
```

In this example, the folder `pharma` is deleted without prompting the user.

Creating a Translation Memory

Purpose

Translation Memorycreating (from command area) creatingTranslation Memory databases (from command area) To set up a new Translation Memory, it might be more convenient to use a command instead of using OpenTM2 windows.

Format

EQFCMDcreating Translation Memory databases commandsEQFCMDcreating Translation Memory databases The following syntax diagram describes how you create a Translation Memory from the Windows^(R) command line. Note that lowercase letters in a parameter name are optional and need not be typed.

**EQFCMD /TAsk=MEMCRT /NAme=memdb /DEsc=desc /TYpe= SHARED
LOCAL /TODrive=drive /SRcIng=source /QUIET**

Parameters

/TAsk=MEMCRT

`MEMCRT` specifies that you want to set up a new Translation Memory.

/NAmemdb

memdb specifies the name of the new Translation Memory.

/DEsc=desc

desc specifies the description of the new Translation Memory (up to 40 characters). If this parameter is not specified, the new Translation Memory will have no description.

/TYpe=

Specifies whether the new Translation Memory is to be shared among several OpenTM2 users (SHARED) or if it is a local one which can be used only by you (LOCAL). If you omit this parameter, LOCAL is assumed.

/TOdrive=drive

drive specifies the drive where the new Translation Memory is to be located.

/SRcLng=source

source is the source language of the text segments stored in the new Translation Memory.

/QUIET

If you specify this parameter, you are not prompted with any message window. If you do not specify this parameter, a message window pops up whenever an error occurs or when the task has successfully completed.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to create a Translation Memory using the EQFCMD command.

Example

```
eqfcmd /task=memcrt /name=medmem /desc="new memory" /type=local  
/todrive=e /srcLng=English(U.S.)
```

In this example, the new Translation Memory *medmem* is a local one and will be located on drive E. The source language of the stored text segments is English (U.S.).

Deleting a Translation Memory

Purpose

Translation Memory deleting (from command area) deleting Translation Memory databases (from command area) If you want to delete a Translation Memory, using a command typed in your system's command area instead of OpenTM2 windows might be quicker and more convenient.

Format

EQFCMD deleting Translation Memory databases commands EQFCMD deleting Translation Memory databases The following syntax diagram describes how you delete a Translation Memory with the EQFCMD command. Note that lowercase letters in a parameter name are optional and need not be typed.

EQFCMD /TAsk=MEMDEL /MEm=memdb /QUIET

Parameters**/TAsk=MEMDEL**

MEMDEL specifies that you want to remove a Translation Memory.

/MEm=memdb

Specifies the name of the Translation Memory that you want to remove.

/QUIET

If you specify this parameter, you are not prompted with any message window. If you do not specify this parameter, a message window pops up whenever an error occurs or when the task has successfully completed.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use

any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to delete a Translation Memory using the EQFCMD command.

Example

```
eqfcmd /task=memdel /mem=medmem
```

In this example, the Translation Memory `medmem` is deleted. You will be prompted with a message window.

Exporting a Translation Memory

Purpose

Translation Memory exporting (from command area) exporting Translation Memory databases (from command area) If you want to export a Translation Memory, using a command typed in your system's command area instead of OpenTM2 windows might be more convenient.

Format

EQFCMD exporting Translation Memory databases commands The following syntax diagram describes how you export a Translation Memory from the Windows^(R) command line. Note that lowercase letters in a parameter name are optional and need not be typed.

EQFCMD /TAsk=MEMEXP /OUT=mem_file /TYpe=EXTERNAL /MEm=memdb /OVerwrite= NO YES /QUIET

Parameters

/TAsk=MEMEXP

`MEMEXP` specifies that you want to export a Translation Memory.

/OUT=mem_file

`mem_file` specifies the name (along with the drive and directory) of a file where the Translation Memory specified in `memdb` is to be exported to. The file must have the extension EXP.

/TYpe=EXTERNAL

`EXTERNAL` specifies that it is an external Translation Memory that you want to export. This parameter is optional.

/MEm=memdb

`memdb` specifies the name of the Translation Memory that you want to export.

/OVerwrite=

Specifies whether an existing Translation Memory file is to be overwritten when exporting the Translation Memory.

/QUIET

If you specify this parameter, you are not prompted with any message window. If you do not specify this parameter, a message window pops up whenever an error occurs or when the task has successfully completed.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to export a Translation Memory using the EQFCMD command.

Example

```
eqfcmd /task=memexp /out=d:\mymemo.exp /type=external /mem=oldmemo
```

In this example, the Translation Memory `oldmemo` is exported to the file `d:\mymemo.exp`.

Importing a Translation Memory

Purpose

Translation Memory importing (from command area) importing Translation Memory databases (from command area) To import a Translation Memory, you might find it more convenient to use a command instead of OpenTM2 windows. You might want to do this if you have many Translation Memory databases that you want to import.

Format

EQFCMD importing Translation Memory databases commands EQFCMD importing Translation Memory databases The following syntax diagram describes how you import a Translation Memory from the command area. Note that lowercase letters in a parameter name are optional and need not be typed.

EQFCMD /TAsk=MEMIMP /FIlles=mem_file /TYpe=EXTERNAL /MEm=memdb /QUIET

Parameters

/TAsk=MEMIMP

MEMIMP specifies that you want to import a Translation Memory.

/FIlles=mem_file

mem_file specifies the name of the Translation Memory that you want to import along with the drive and directory where it is located.

/TYpe=EXTERNAL

EXTERNAL specifies that it is an external Translation Memory (in SGML format) that you want to import. This parameter is optional.

/MEm=memdb

memdb specifies the name of an existing Translation Memory to which the Translation Memory specified in *mem_file* is to be imported to. The contents of these Translation Memory databases will then be merged.

/QUIET

If you specify this parameter, you are not prompted with any message window. If you do not specify this parameter, a message window pops up whenever an error occurs or when the task has successfully completed.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to import a Translation Memory using the EQFCMD command.

Example

```
eqfcmd /task=memimp /files=d:\mymemo.mem /type=external /mem=oldmemo
```

In this example, the Translation Memory *mymemo.mem*, which is in external format, is imported to the Translation Memory *oldmemo*.

Exporting a dictionary

Purpose

dictionary exporting (from command area) exporting dictionaries (from command area) If you want to export a dictionary, using a command typed in your system's command area instead of OpenTM2 windows might be more convenient.

Format

EQFCMD exporting dictionaries commands EQFCMD exporting dictionaries The following syntax diagram describes how you export a Translation Memory from the Windows^(R) command line. Note that lowercase letters in a parameter name are optional and need not be typed.

EQFCMD /TAsk=DICEXP /Dlct=dic_name /OUT=mem_file /OVerwrite= NO YES /QUIET

Parameters

/TAsk=DICEXP

DICEXP specifies that you want to export a dictionary.

/Dlct=dic_name

dic_file specifies the name of the dictionary to be exported.

/OUT=mem_file

mem_file specifies the name (along with the drive and directory) of a file where the dictionary is to be exported to. The file must have the extension EXP.

/OVerwrite=

Specifies whether an existing dictionary file is to be overwritten with the exported dictionary.

/QUIET

If you specify this parameter, you are not prompted with any message window. If you do not specify this parameter, a message window pops up whenever an error occurs or when the task has successfully completed.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to export a dictionary using the EQFCMD command.

Example

```
eqfcmd /task=dicexp /dict=mydict /out=d:\mydict.exp
```

In this example, the dictionary `mydict` is exported to the file `d:\mydict.exp`.

Importing a dictionary

Purpose

dictionaryimporting (from command area) importingdictionaries (from command area) If you want to import a dictionary, using a command typed in your system's command area instead of OpenTM2 windows might be more convenient.

Format

EQFCMDimporting dictionaries commandsEQFCMDimporting dictionaries The following syntax diagram describes how you import a dictionary from the Windows^(R) command area. Note that lowercase letters in a parameter name are optional and need not be typed.

EQFCMD /TAsk=DICIMP /FIlles=dict_file /Dlct=dict_name /OPtions= COMBINE REPLACE IGNORE /PAssword=password /QUIET

Parameters

/TAsk=DICIMP

DICIMP specifies that you want to import a dictionary.

/FIlles=dict_file

Specifies the name of the SGML file containing the dictionary that you want to import along with the drive and directory where it is located.

/Dlct=dict_name

Specifies the name of an existing dictionary to which the dictionary specified in `FIlles` is to be imported to. It can be a local or a shared dictionary.

/OPtions

This parameter is particularly useful if the existing dictionary already contains entries. It specifies the following:

- **REPLACE**
The entries in the SGML (external) dictionary file replace the entries in the existing (internal) dictionary.
- **IGNORE**

- **IGNORE**
Ignores the entries in the SGML (external) dictionary.
- **COMBINE**
Combines the entries in the SGML (external) dictionary file with the entries in the existing (internal) dictionary. This is the default. That means if nothing is specified, the system assumes that you want to combine the dictionary entries.

/PPassword

password specifies the password of the dictionary to which the SGML file is to be imported (if this dictionary is write-protected). If the dictionary is write-protected and no password is entered, processing stops.

/QUIET

If you specify this parameter, you are not prompted with any message window. If you do not specify this parameter, a message window pops up whenever an error occurs or when the task has successfully completed.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to import a dictionary using the EQFCMD command.

Example

```
eqfcmd /task=dicimp /files=d:\mymemo.sgm /options=replace /dict=newmemo
```

In this example, the dictionary file *mymemo.sgm*, which is in external (SGML) format, is imported to the existing dictionary *newmemo*. The entries in *mymemo.sgm* replace the entries in *newmemo*.

Archiving a Translation Memory

Purpose

archiving Translation Memory databases from command area Translation Memoryarchiving (from command area) To archive a **Translation Memory** it might be more convenient to use a command instead of **OpenTM2** windows.

Format

EQFCMDarchiving Translation Memory databases commandsEQFCMDarchiving Translation Memory databases The following syntax diagram describes how you archive a Translation Memory with the EQFBATCH command. Note that lowercase letters in a parameter name are optional and need not be typed.

EQFBATCH /TAsk=ARCHTM /FLD=folder /FIlles= document_name (,document_name) @list_file_name /MEm=memdb /OVerwrite= NO YES

Parameters

/TAsk=ARCHTM

ARCHTM specifies that you want to archive a **Translation Memory**.

/FLD=folder

folder specifies the name of the folder that contains the documents of which you want to archive the translated segments.

/FIlles=document_name

document_name specifies the name of a document to be archived. The *document_name* specification can include the drive and directory containing the document to be archived.

If you specify several documents, enclose the document names in brackets and separate them by commas. If you want to specify a list of document names in a list file, refer to this list file by preceding the list file name with the @ symbol (@ *list_file_name*). In the list file enclose the document names in brackets and separate them by commas.

If you do not specify the **FIlles** option, all documents in the folder are archived.

/MEm=memdb

memdb specifies the name of the Translation Memory to be archived.

/OVerwrite=

Specifies if an existing **Translation Memory** is to be overwritten when archiving starts (NO is the default).

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to archive a **Translation Memory** using the EQFBATCH command.

Example

```
eqfbatch /task=archtm /fld=test /mem=arch
```

In this example, the folder test is archived in **Translation Memory** arch.

Organizing a Translation Memory

Purpose

organizing Translation Memory databases (from command area) To organize a **Translation Memory** it might be more convenient to use a command instead of OpenTM2 windows. It is particularly useful if you want to regularly organize several or all **Translation Memory databases** overnight.

Format

EQFCMDorganizing Translation Memory databases commands The following syntax diagram describes how you organize a Translation Memory with the EQFCMD command. Note that lowercase letters in a parameter name are optional and need not be typed.

EQFCMD /TAsk=MEMORG /MEm= memdb (,memdb) @list_member_name

Parameters

/TAsk=MEMORG

MEMORG specifies that you want to organize a **Translation Memory**.

/MEm=memdb

memdb specifies the name of the Translation Memory to be organized.

If you want to specify several **Translation Memory databases**, enclose the Translation Memory database names in brackets and separate them by commas.

If you want to specify a list of Translation Memory database names in a list file, refer to this list file by preceding the list file name with the @ symbol (@ list_member_name). In the list file enclose the Translation Memory database names in brackets and separate them by commas.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to organize a **Translation Memory** using the EQFCMD command.

Example

```
eqfcmd /task=memorg /mem=sample1
```

In this example, the Translation Memory sample1 is organized.

Opening a document

Purpose

documentopening (from command area) Instead of using OpenTM2 windows to open a document, you might find it more convenient to use a command.

Format

EQFCMD opening documents commands
EQFCMD opening documents The following syntax diagram describes how you open a document from the Windows^(R) command line. Note that lowercase letters in a parameter name are optional and need not be typed.

EQFCMD
/TAsk=DOCOPEN /FLD=folder /NAme=document_name /SEgment=segnumber /QUIET

Parameters

/TAsk=DOCOPEN

DOCOPEN specifies that you want to open a document.

/FLD=folder

folder specifies the name of the folder that contains the document you want to open.

/NAme=document_name

document_name specifies the name of the document to be opened.

/SEgment=segnumber

segnumber specifies the number of the segment which is to be activated when the document is opened.

/QUIET

If you specify this parameter, you are not prompted with any message window. If you do not specify this parameter, a message window pops up whenever an error occurs or when the task has successfully completed.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to open a document using the EQFCMD command.

Example

```
eqfcmd /task=docopen /fld=sample1 /name=device.scr
```

This command opens the document DEVICE.SCR which is stored in folder SAMPLE1.

Counting words

Purpose

countingwords (from command area) Translators are usually paid for translated words or lines. To prepare the payment it is necessary to count the words that have been translated for a specific project. Also, to plan for a translation project it is necessary to know how many words will have to be translated.

Instead of using OpenTM2 windows to count words in a document, using a command might be more convenient.

Format

EQFCMD counting words commands
EQFCMD counting words The following syntax diagram describes how you start to count words from the Windows^(R) command area. Note that lowercase letters in a parameter name are optional and need not be typed.

EQFCMD /TAsk=WORDCNT /FLD=folder /FIles= document_name (,document_name) @list_file_name /OUT=cnt_name /OVerwrite= NO YES /OOptions= SOURCE TMMATCH TARGET /QUIET

Parameters

/TAsk=WORDCNT

WORDCNT specifies that you want to count words in a document.

/FLD=folder

folder specifies the name of the folder that contains the document of which you want to count words.

/Files=*document_name*

document_name specifies the name of the document of which you want to count words.

If you specify several documents, enclose the document names in brackets and separate them by commas.

If you want to specify a list of document names in a list file, refer to this list file by preceding the list file name with the @ symbol (@ *list_file_name*). In the list file enclose the document names in brackets and separate them by commas.

If you do not specify the **Files** option, the words of all documents in the folder are counted.

/OUT=*cnt_name*

cnt_name specifies the name of the file where the count result is to be stored (along with the drive and directory information).

/OVerwrite=

Specifies whether the file specified in *cnt_name* is to be overwritten if it already exists (NO is the default).

/OPtions=*option*

option specifies whether you want to count the words in the translated document (TARGET) or in the original document (SOURCE), or whether you want to count the number and type of matches found in the **Translation Memory** (TMMATCH). SOURCE is the default.

/QUIET

If you specify this parameter, you are not prompted with any message window. If you do not specify this parameter, a message window pops up whenever an error occurs or when the task has successfully completed.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to count words using the EQFCMD command.

Example

```
eqfcmd /task=wordcnt /fld=newfld /files=161.scr /options=target  
/out=e:\161.cnt
```

All words in the translated document 161.scr, which is contained in folder newfld, are counted and the result is stored in the file 161.cnt on drive E.

Creating reports

Purpose

creatingreports (from command area) OpenTM2 enables you to collect information about the effort, state, and history of your translations. The collected information can be displayed in different reports and used as a basis for calculating translation costs. Instead of using OpenTM2 windows to create reports, using a command might be more convenient.

Format

EQFCMDcreating reports commandsEQFCMDcreating reports The following syntax diagram describes how you start to create reports from the Windows (R) command area. Note that lowercase letters in a parameter name are optional and need not be typed.

```
EQFCMD /Task=CNTRPT /FLD=folder /Files= document_name (,document_name )  
@list_file_name /OUT=output_name /Report= HISTORY COUNTING CALCULATING  
PREANALYSIS REDUNDANCY REDSEGLIST /TYpe= DATE BRIEF DETAIL WITH_TOTALS  
WITHOUT_TOTALS BASE BASE_SUMMARY BASE_SUMMARY_FACT SUMMARY_FACT  
FACT /Profile=profile /OVerwrite= NO YES /Format= XML ASCII HTML /QUIET
```

Parameters

/Task=CNTRPT

CNTRPT specifies that you want to create a report.

/FLD=*folder*

folder specifies the name of the folder that contains the documents of which you want to create a report.

/Files=*document_name*

document_name specifies the name of the document of which you want to create a report.

If you specify several documents, enclose the document names in brackets and separate them by commas.

If you want to specify a list of document names in a list file, refer to this list file by preceding the list file name with the @ symbol (@ *list_file_name*). In the list file enclose the document names in brackets and separate them by commas.

If you do not specify the **Files** option, all documents in the folder are selected.

/OUT=*output_name*

output_name specifies the name of the file where the report is to be stored (along with the drive and directory information).

/REport=*rpt_name*

rpt_name specifies whether you want to create a History, Counting, Calculating, Preanalysis, or Redundancy Report or a Redundant Segment List.

/Type=*rpt_type*

rpt_type specifies the type of report.

The following types are available for a History Report:

- DATE
To get a brief report that is sorted by the creation date and time of the documents.
- BRIEF
To get a brief report where the information is sorted by document name.
- DETAIL
To get a detailed report of the process tasks performed on the selected documents.

The following types are available for a Counting Report:

- WITH_TOTALS
To get a report on one or more documents in the folder, with a summary at its end.
- WITHOUT_TOTALS
To get a report on one or more documents in the folder, without a summary.

The following types are available for a Calculating, Preanalysis, or Redundancy Report:

- BASE
To get a rough overview, for each selected document, of the activities performed by OpenTM2 and by the translator to complete a translation.
- BASE_SUMMARY
To get an overview and a summary of the activities performed for selected documents.
- FACT
To get a detailed overview of the costs of a translation using different complexity and pay factors.
- BASE_SUMMARY_FACT
To get a report containing all the information previously described.
- SUMMARY_FACT
To get a report containing both the summary and the fact sheet.

/Profile=*profile*

profile specifies the name of the profile to be loaded. When this parameter is omitted, the profile recently used for the folder is used. To suppress the usage of profiles specify NONE as profile name.

/Overwrite=

Specifies whether the file specified in *output_name* is to be overwritten if it already exists (NO is the default).

/Format=

Specifies the output format for the report. The following formats can be used:

- ASCII
To create the report in plain ASCII text
- XML

- To create the report in the XML format
- HTML
 - To create the report in HTML format

/QUIET

If you specify this parameter, you are not prompted with any message window. If you do not specify this parameter, a message window pops up whenever an error occurs or when the task has successfully completed.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to create a report using the EQFCMD command.

Example

```
eqfcmd /task=cntrpt /fld=sample1 /out=output /report=history /type=date
```

In this example, a History Report of folder sample1 is created and stored in the file output.

Performing several tasks in one step

Purpose

Instead of typing all parameters in one line, you can use the /TASKLIST option and specify the parameters for any task in a separate file.

Format

EQFCMD /TASKLIST=*list_of_tasks* /QUIET

Parameters

/TASKLIST=*list_of_tasks*

list_of_tasks is the name of a file containing a task description on each line, for example, parameters for several import or export requests (see examples).

/QUIET

Can only be specified with the /TASKLIST option to suppress messages. If this option is omitted, a message window pops up whenever an error occurs or when a task has successfully completed.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to specify the export parameters in a separate file (taskexp.fil) to export several documents (med1.txt and med2.txt) and then call the EQFCMD command with a reference to this file.

taskexp.fil

```
/task=docexp /fld=pharma /files=c:\med1.txt /options=target  
/overwrite=yes  
/task=docexp /fld=pharma /files=c:\med2.txt /options=target /overwrite=yes
```

Enter the EQFCMD command as follows:

Example

```
eqfcmd /tasklist=taskexp.fil
```

This lets you export more than one document with one command. All EQFCMD tasks can be combined.

The next example shows how to specify the import parameters in a separate file (taskimp.fil) to import several documents listed in two files (@meddoca.lst and @meddocb.lst) and then call the EQFCMD command with a reference to this file.

taskimp.fil

```
/task=docimp /fld=pharma /files=@meddoca.lst /overwrite=no  
/task=docimp /fld=pharma /files=@meddocb.lst /overwrite=no
```

Enter the EQFCMD command as follows:

Example

```
eqfcmd /tasklist=taskimp.fil
```

This lets you import more than one document with one command. All EQFCMD tasks can be combined.

Working from the command area (EQFBATCH)

Using the OpenTM2 windows is only one way of performing translation tasks with this product. An alternative way is to type instructions directly in the system's command area.

To start a task from the command area:

1. Type a command and its parameters in your operating system's command area (next to the command prompt) according to the command syntax described in the following chapters.
2. Press Enter.

Note that lowercase letters in a parameter name are optional and need not be typed. A mixture between uppercase and lowercase letters is allowed. However, a muddled letter sequence like in OVerWRiT e must not be used. The parameters can be entered in any order whatever.

To process several commands in sequence, type the individual commands into a batch file (*.BAT or *.CMD) using any text editor. You can execute these commands by starting the batch file.

To avoid retyping of long text strings in commands, you can also type a command and its parameters in a text editor, which is capable of the **Copy to clipboard** function. You can then **paste** the text string from the clipboard directly into your system's command area and start the task.

Why use EQFBATCH instead of EQFCMD

- For EQFBATCH no running OpenTM2 session is required.
- EQFBATCH is faster than EQFCMD as the OpenTM2 functions are called directly.
- New OpenTM2 functions (e.g. export documents in validation format) will only be available in EQFBATCH

How to read syntax diagrams

In this chapter diagrams are used to illustrate the programming syntax. To use a diagram, follow a path from left to right, top to bottom, adding elements as you go. In these diagrams, all spaces and other characters are significant.

Each diagram begins with a double right arrowhead and ends with a right and left arrowhead pair. Lines beginning with single right arrowheads are continuation lines.

keyword=variable_value

Keywords are all in lowercase, but can be entered in uppercase or in lowercase. Variable values that you provide are shown in *italics* and are usually in lowercase. Where values are shown in uppercase, they should be entered as they appear.

In a choice of items, the default item is always shown above the main line:

keyword= default_value other_value other_value

Optional syntax elements are shown below the main line:

keyword=value

A repeat arrow shown above an item or a stack of items indicates that you can specify the item multiple times or specify more than one of the items. A character (such as a comma) on the repeat arrow line indicates that the items must be separated by that character. A repeatable operand is shown like this:

keyword= (,variable_value)

Syntax diagrams can be broken into fragments. A fragment is indicated by vertical bars with the name of the fragment between the bars. The fragment is shown following the main diagram, like so:

/KEYWORD= ITEM1 variable1 variable2 variable3

Analyzing a document or folder

Purpose

documentEQFBATCH: analyzing (from command area) documents (from command area) To analyze a document or a set of documents it might be more convenient to use a command instead of OpenTM2 windows. It is particularly useful if you have many large documents or folders that you want to analyze overnight.

Format

EQFBATCHAnalyzing documents commandsEQFBATCHAnalyzing a document or folder The following syntax diagram describes how you analyze documents from the Windows^(R) command line. Note that lowercase letters in a parameter name are optional and need not be typed.

```
EQFBATCH /TAsk=ANALYSIS /FLD=folder /FIlles= document_name (,document_name )
@list_file_name /OOptions=(,) /MEm= memdb2 (,memdb ) /OVerwrite= NO
YES /QUIET /QUIET=NOMSG option ADDTOMEM AUTO AUTOCONTEXT AUTOLAST
AUTOJOIN TMMATCH UNTRANSLATED ADJUSTLEADWS ADJUSTTRAILWS
REDUNDCOUNT STOPATFIRSTEXACT IGNORECOMMENTED RESPECTCRLF
```

Parameters

/TAsk=ANALYSIS

ANALYSIS specifies that you want to analyze documents.

/FLD=folder

folder specifies the name of a folder that contains the documents that you want to analyze.

/FIlles=document_name

document_name specifies the name of a document to be analyzed.

If you specify several documents, enclose the document names in brackets and separate them by commas.

If you want to specify a list of document names in a list file, refer to this list file by preceding the list file name with the @ symbol (@ list_file_name). In the list file enclose the document names in brackets and separate them by commas.

If you do not specify the FIlles option, all documents in the folder are analyzed.

/OOptions=

This parameter is particularly useful if you have Translation Memory databases available from previous translations. It specifies the following:

- ADDTOMEM
OpenTM2 can search existing Translation Memory databases for segments that match segments in the document being analyzed. Segments that match exactly or almost exactly are copied into the Translation Memory associated to the document or folder being analyzed. In that way, you can start to fill a document-specific Translation Memory even before its translation has begun.
- AUTO
You can choose to have Translation Manager automatically replace sentences with matches that already exist in the Translation Memory. This substitution is done only for exact matches. The document is then already partially translated when you start working with it, enabling you to focus on the translation of new segments.
- AUTOCONTEXT
See the explanation for AUTO. However, if more than one exact match has been found, the translation of the segment is to be taken from the same document.
- AUTOLAST
See the explanation for AUTO. However, if more than one exact match has been found, the most recent translation of the segment is to be taken.
- AUTOJOIN
Joins neighboring segments. This can be useful if you want to analyze a document that you have worked with before and where you joined neighboring text segments before you translated them. During a new analysis run these segments are treated as separate units again. However, in the associated Translation Memory there is a match (translation) for the joined segments only. To find this match more easily during automatic substitution, this option joins neighboring segments in the document to be analyzed. Note that neighboring segments are only joined if for the first segment a fuzzy match has been found.
- TMMATCH

2 Mandatory only when OOptions=ADDTOMEM

OpenTM2 counts the number and type of matches found in the **Translation Memory** for the document. The document is then segmented and the matches are counted, but no segments are replaced with matches found in the **Translation Memory**

If you also specify **AUTO**, **OpenTM2** automatically replaces segments with matches found in the **Translation Memory** and counts the number and type of matches replaced.

- **UNTRANSLATED**
All untranslated segments are to be stored in a separate file.
- **ADJUSTLEADWS**
During whitespace handling of automatic substitution, leading whitespaces are adjusted to whitespaces in source segment.
- **ADJUSTTRAILWS**
During whitespace handling of automatic substitution, trailing whitespaces are adjusted to whitespaces in source segment.
- **RESPECTCRLF**
During whitespace handling of automatic substitution, substitution takes place with respect of CRLF in the segment.
- **REDUNTCOUNT**
Prepare information for redundancy counting
- **STOPATFIRSTEXACT**
Do no look for memory proposals in the remaining memories when an exact match has been found
- **IGNORERECOMMENDED**
Ignore memory proposals which have a comment

/MEm=memdb

memdb specifies a Translation Memory or a list of Translation Memory databases that is to be searched when **ADDTOMEM** has been specified.

/Overwrite=

Specifies if an existing document will be overwritten when its translation has already been started. (**NO** is the default).

/QUIET

If you specify the parameter in this format, you are not prompted with any message window. If you do not specify any of the /quiet parameters, a message window pops up whenever an error occurs or when the task has successfully completed.

/QUIET=NOMSG

If you specify this parameter, you are not prompted with any message window. Any error message text is routed to stdout.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to analyze a document by using the EQFBATCH command.

Example

```
eqfbatch /task=analysis /fld=myfol /files=16.scr  
/options=(addtomem,auto,untranslated) /mem=oldmem
```

In this example, the document to be analyzed has the file name **16.scr** and is stored in folder **myfol**. Any matches found in the Translation Memory **oldmem** are to be added to the Translation Memory of folder **myfol**. Any original segment for which an exact match has been found is to be replaced with its corresponding translation segment. All the remaining segments that are not translated are to be stored in a separate file.

Archiving a Translation Memory

Purpose

archiving **Translation Memory databases** from command area (EQFBATCH) Translation MemoryEQFBATCH: archiving (from command area) To archive a **Translation Memory** it might be more convenient to use a command instead of **OpenTM2** windows.

Format

EQFBATCHarchiving **Translation Memory databases** commandsEQFBATCHarchiving **Translation Memory databases** The following syntax diagram describes how you archive a

Translation Memory with the EQFBATCH command. Note that lowercase letters in a parameter name are optional and need not be typed.

EQFBATCH /TAsk=ARCHTM /FLD=folder /FIlEs= document_name (,document_name) @list_file_name /MEm=memdb /OVerwrite= NO YES /OPtions=options

Parameters

/TAsk=ARCHTM

ARCHTM specifies that you want to archive a **Translation Memory**.

/FLD=folder

folder specifies the name of the folder that contains the documents of which you want to archive the translated segments.

/FIlEs=document_name

document_name specifies the name of a document to be archived. The document_name specification can include the drive and directory containing the document to be archived.

If you specify several documents, enclose the document names in brackets and separate them by commas.

If you want to specify a list of document names in a list file, refer to this list file by preceding the list file name with the @ symbol (@ list_file_name). In the list file enclose the document names in brackets and separate them by commas.

If you do not specify the FIlEs option, all documents in the folder are archived.

/MEm=memdb

memdb specifies the name of the Translation Memory to be archived.

/OVerwrite=

Specifies if an existing **Translation Memory** is to be overwritten when archiving starts (NO is the default).

/OPtions=options

option can be one of the following: USEASFOLDERTM to use the archive memory as folder memory SOURCESOURCEMEM to create a source-source memory using the untranslated segments SETMFLAG to set the machine translation flag of the segments in the archive memory

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to archive a **Translation Memory** using the EQFBATCH command.

Example

eqfbatch /task=archtm /fld=test /mem=arch

In this example, the folder test is archived in **Translation Memory** arch.

Creating reports

Purpose

creatingEQFBATCH: reports (from command area) OpenTM2 enables you to collect information about the effort, state, and history of your translations. The collected information can be displayed in different reports and used as a basis for calculating translation costs. Instead of using OpenTM2 windows to create reports, using a command might be more convenient.

Format

EQFBATCHcreating reports commandsEQFBATCHcreating reports The following syntax diagram describes how you start to create reports from the Windows^(R) command area. Note that lowercase letters in a parameter name are optional and need not be typed.

EQFBATCH /TAsk=CNTRPT /FLD=folder /FIlEs= document_name (,document_name) @list_file_name /OUT=output_name /REport= HISTORY COUNTING CALCULATING PREANALYSIS REDUNDANCY REDSEGLIST /TYpe= DATE BRIEF DETAIL WITH_TOTALS

**WITHOUT_TOTALS BASE BASE_SUMMARY BASE_SUMMARY_FACT SUMMARY_FACT
FACT /PProfile=profile /OVerwrite= NO YES /FOrmat= XML ASCII
HTML /QUIET /QUIET=NOMSG**

Parameters

/TAsk=CNTRPT

CNTRPT specifies that you want to create a report.

/FLD=folder

folder specifies the name of the folder that contains the documents of which you want to create a report.

/FIlEs=document_name

document_name specifies the name of the document of which you want to create a report.

If you specify several documents, enclose the document names in brackets and separate them by commas.

If you want to specify a list of document names in a list file, refer to this list file by preceding the list file name with the @ symbol (@ list_file_name). In the list file enclose the document names in brackets and separate them by commas.

If you do not specify the FIlEs option, all documents in the folder are selected.

/OUT=output_name

output_name specifies the name of the file where the report is to be stored (along with the drive and directory information).

/REport=rpt_name

rpt_name specifies whether you want to create a History, Counting, Calculating, Preanalysis, or Redundancy Report or a Redundant Segment List.

/TYpe=rpt_type

rpt_type specifies the type of report. The following types are available for a History Report:

- DATE
To get a brief report that is sorted by the creation date and time of the documents.
- BRIEF
To get a brief report where the information is sorted by document name.
- DETAIL
To get a detailed report of the process tasks performed on the selected documents.

The following types are available for a Counting Report:

- WITH_TOTALS
To get a report on one or more documents in the folder, with a summary at its end.
- WITHOUT_TOTALS
To get a report on one or more documents in the folder, without a summary.

The following types are available for a Calculating, Preanalysis, or Redundancy Report:

- BASE
To get a rough overview, for each selected document, of the activities performed by OpenTM2 and by the translator to complete a translation.
- BASE_SUMMARY
To get an overview and a summary of the activities performed for selected documents.
- FACT
To get a detailed overview of the costs of a translation using different complexity and pay factors.
- BASE_SUMMARY_FACT
To get a report containing all the information previously described.
- SUMMARY_FACT
To get a report containing both the summary and the fact sheet.

/PProfile=profile

profile specifies the name of the profile to be loaded.

/OVerwrite=

Specifies whether the file specified in output_name is to be overwritten if it already exists (NO is the default).

/Format=

Specifies the output format for the report. The following formats can be used:

- ASCII
To create the report in plain ASCII text
- XML
To create the report in the XML format
- HTML
To create the report in HTML format

/QUIET

If you specify the parameter in this format, you are not prompted with any message window. If you do not specify any of the /quiet parameters, a message window pops up whenever an error occurs or when the task has successfully completed.

/QUIET=NOMSG

If you specify this parameter, you are not prompted with any message window. Any error message text is routed to stdout.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to create a report using the EQFBATCH command.

Example

```
eqfbatch /task=cntrpt /fld=sample1 /out=output /report=history /type=date
```

In this example, a History Report of folder sample1 is created and stored in the file output.

Counting words

Purpose

countingEQFBATCH: words (from command area) Translators are usually paid for translated words or lines. To prepare the payment it is necessary to count the words that have been translated for a specific project. Also, to plan for a translation project it is necessary to know how many words will have to be translated.

Instead of using OpenTM2 windows to count words in a document, using a command might be more convenient.

Format

EQFBATCHcounting words commandsEQFBATCHcounting words The following syntax diagram describes how you start to count words from the Windows (R) command area. Note that lowercase letters in a parameter name are optional and need not be typed.

```
EQFBATCH /TAsk=WORDCNT /FLD=folder /FIlles= document_name ( document_name )
@list_file_name /OUT=cnt_name /OVerwrite= NO YES /OPtions= SOURCE TMMATCH
TMMATCH, SEPREPLMATCH) TARGET DUPLICATE DUPMEMMATCH
FUZZYMATCH /FOrmat= ASCII HTML XML /QUIET /QUIET=NOMSG
```

Parameters

/TAsk=WORDCNT

WORDCNT specifies that you want to count words in a document.

/FLD=folder

folder specifies the name of the folder that contains the document of which you want to count words.

/FIlles=document_name

document_name specifies the name of the document of which you want to count words.

If you specify several documents, enclose the document names in brackets and separate them by commas.

If you want to specify a list of document names in a list file, refer to this list file by preceding the list file name with the @ symbol (@ list_file_name). In the list file enclose the document names in

brackets and separate them by commas.

If you do not specify the **Files** option, the words of all documents in the folder are counted.

/OUT=cnt_name

cnt_name specifies the name of the file where the count result is to be stored (along with the drive and directory information).

/OVerwrite=

Specifies whether the file specified in *cnt_name* is to be overwritten if it already exists (**NO** is the default).

/OPtions=option

option specifies whether you want to count the words

- in the translated document (**TARGET**)
- or in the original document (**SOURCE**)
- or the duplicate words (**DUPLICATE**)
- or duplicate words with added memory match information (**DUPMEMMATCH**)
- or fuzzy matches per class (**FUZZYMATCH**)
- or whether you want to count the number and type of matches found in the **Translation Memory** (**TMMATCH**)
- or use **TMMATCH** together with **SEPREPLMATCH** to count replaced matches separately. **SOURCE** is the default.

/FOrmat=format

format specifies the output format for the word count results and can be ASCII (= plain text), HTML, or XML

/QUIET

If you specify the parameter in this format, you are not prompted with any message window. If you do not specify any of the /quiet parameters, a message window pops up whenever an error occurs or when the task has successfully completed.

/QUIET=NOMSG

If you specify this parameter, you are not prompted with any message window. Any error message text is routed to stdout.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to count words using the EQFBATCH command.

Example

```
eqfbatch /task=wordcnt /fld=newfld /files=161.scr /options=target  
/out=e:\161.cnt
```

All words in the translated document **161.scr**, which is contained in folder **newfld**, are counted and the result is stored in the file **161.cnt** on drive E.

Exporting a dictionary

Purpose

dictionaryEQFBATCH: exporting (from command area) exportingEQFBATCH: dictionaries (from command area) If you want to export a dictionary, using a command typed in your system's command area instead of OpenTM2 windows might be more convenient.

Format

EQFBATCHexporting dictionaries commandsEQFBATCHexporting dictionaries The following syntax diagram describes how you export a Translation Memory from the Windows^(R) command line. Note that lowercase letters in a parameter name are optional and need not be typed.

**EQFBATCH /TAsk=DICEXP /DDict=dic_name /OUT=mem_file /OVerwrite= NO
YES /OPTIONS=option /QUIET /QUIET=NOMSG**

Parameters

/TAsk=DICEXP

DICEXP specifies that you want to export a dictionary.

/DIct=*dict_name*

dict_file specifies the name of the dictionary to be exported.

/OUT=*mem_file*

mem_file specifies the name (along with the drive and directory) of a file where the dictionary is to be exported to. The file must have the extension EXP.

/OVerwrite=

Specifies whether an existing dictionary file is to be overwritten with the exported dictionary.

/OPTIONS=*option*

Specifies the format of the exported dictionary. Use ASCII for ASCII encoding, ANSI for Ansi encoding or UTF16 for Unicode (UTF-16) encoding of the exported dictionary. If nothing is specified the dictionary is exported in the ASCII format.

/QUIET

If you specify the parameter in this format, you are not prompted with any message window. If you do not specify any of the /quiet parameters, a message window pops up whenever an error occurs or when the task has successfully completed.

/QUIET=NOMSG

If you specify this parameter, you are not prompted with any message window. Any error message text is routed to stdout.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to export a dictionary using the EQFBATCH command.

Example

```
eqfbatch /task=dicexp /dict=mydict /out=d:\mydict.exp
```

In this example, the dictionary `mydict` is exported to the file `d:\mydict.exp`.

Importing a dictionary

Purpose

EQFBATCH: importing (from command area) dictionaries (from command area) If you want to import a dictionary, using a command typed in your system's command area instead of OpenTM2 windows might be more convenient.

Format

EQFBATCH: importing dictionaries commands EQFBATCH: importing dictionaries The following syntax diagram describes how you import a dictionary from the Windows (R) command area. Note that lowercase letters in a parameter name are optional and need not be typed.

EQFBATCH /TAsk=DICIMP /FIles=*dict_file* **/DIct=***dict_name* **/OPtions=** COMBINE REPLACE IGNORE **/OPtions=***options* **/PAssword=***password* **/QUIET** **/QUIET=NOMSG**

Parameters

/TAsk=DICIMP

DICIMP specifies that you want to import a dictionary.

/FIles=*dict_file*

Specifies the name of the SGML file containing the dictionary that you want to import along with the drive and directory where it is located.

/DIct=*dict_name*

Specifies the name of an existing dictionary to which the dictionary specified in `FIles` is to be imported to. It can be a local or a shared dictionary.

OPtions

This parameter controls the format of the imported dictionary and the handling for entries which already exist. For the dictionary form use one of the following values:

- ANSI
For external dictionaries in the Ansi format
- ASCII
For external dictionaries in the ASCII format
- UTF16
For external dictionaries in Unicode (UTF-16) format

For the handling of existing dictionary entries use one of the following values:

- REPLACE
The entries in the SGML (external) dictionary file replace the entries in the existing (internal) dictionary.
- IGNORE
Ignores the entries in the SGML (external) dictionary.
- COMBINE
Combines the entries in the SGML (external) dictionary file with the entries in the existing (internal) dictionary. This is the default. That means if nothing is specified, the system assumes that you want to combine the dictionary entries.

If you specify more than one value you have to enclose the values in parenthesis and separate them using the comma; e.g. **/OPTIONS=(ANSI,REPLACE)** to import a dictionary in Ansi format and replace existing dictionary entries with the imported entries.

/PAssword

`password` specifies the password of the dictionary to which the SGML file is to be imported (if this dictionary is write-protected). If the dictionary is write-protected and no password is entered, processing stops.

/QUIET

If you specify the parameter in this format, you are not prompted with any message window. If you do not specify any of the /quiet parameters, a message window pops up whenever an error occurs or when the task has successfully completed.

/QUIET=NOMSG

If you specify this parameter, you are not prompted with any message window. Any error message text is routed to stdout.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to import a dictionary using the EQFBATCH command.

Example

```
eqfbatch /task=dicimp /files=d:\mymemo.sgm /options=replace /dict=newmemo
```

In this example, the dictionary file `mymemo.sgm`, which is in external (SGML) format, is imported to the existing dictionary `newmemo`. The entries in `mymemo.sgm` replace the entries in `newmemo`.

Deleting documents

Purpose

documentEQFBATCH: deleting (from command area) deletingEQFBATCH: documents (from command area) If you want to delete a document, using a command typed in your system's command area instead of OpenTM2 windows might be quicker and more convenient.

Format

EQFBATCHdeleting documents commandsEQFBATCHdeleting documents The following syntax diagram describes how you delete documents with the EQFBATCH command. Note that lowercase letters in a parameter name are optional and need not be typed.

**EQFBATCH /TAsk=DOCDEL /FLD=folder /FIles= file_name (,file_name)
@list_file_name /QUIET /QUIET=NOMSG**

Parameters

/TAsk=DOCDEL

DOCDEL specifies that you want to remove documents.

/FLD=folder

Specifies the name of the folder that contains the documents to be removed.

/FIlEs=file_name

file_name specifies the name of the document you want to remove. The *file_name* specification can include the drive and directory where the file to be removed is found.

If you specify several documents, enclose the document names in brackets and separate them by commas.

If you want to specify a list of document names in a list file, refer to this list file by preceding the list file name with the @ symbol (@ *list_file_name*). In the list file enclose the document names in brackets and separate them by commas.

/QUIET

If you specify the parameter in this format, you are not prompted with any message window. If you do not specify any of the /quiet parameters, a message window pops up whenever an error occurs or when the task has successfully completed.

/QUIET=NOMSG

If you specify this parameter, you are not prompted with any message window. Any error message text is routed to stdout.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to delete documents using the EQFBATCH command.

Example

```
eqfbatch /task=docdel /fld=pharma /files=(med.txt) /quiet
```

In this example, the document med.txt is removed from folder pharma without prompting the user.

Exporting documents

Purpose

documentEQFBATCH: exporting (from command area) exportingEQFBATCH: documents (from command area) If you have many large documents that you want to export, using the command area of your operating system instead of OpenTM2 windows might be more convenient.

Before exporting, ensure that:

- **OpenTM2** has been started.
- A folder exists that contains the documents to be exported.

The properties of the folder or of the documents give information on the used markup table and other document-related information that is necessary when exporting the respective documents.

To work from the command area, use the **EQFBATCH** command. Its syntax is described in the following chapter.

Format

EQFBATCH exporting documents commandsEQFBATCH exporting documents The following syntax diagrams describe how you export a document from the Windows (R) command area. Note that lowercase letters in a parameter name are optional and need not be typed.

EQFBATCH /TAsk=DOCEXP /FLD=folder /SStartpath=startpath /FIlEs= file_name (,file_name) @list_file_name /OOptions= TARGET SOURCE SNOMATCH VALDOC VAL = valoptions /OVerwrite= NO YES /QUIET /QUIET=NOMSG

Parameters

/Task=DOCEXP

DOCEXP specifies that you want to export a document.

/FLD=*folder*

folder specifies the name of the folder containing the documents that you want to export.

/Startpath=*startpath*

startpath specifies the drive, directory, or both where the file exported is to be placed. The path information specified becomes part of, that is it prefixes, the name of the document. The directory specified must already exist.

If you omit this parameter, the document is placed in the drive and directory specified in *file_name*. However, this drive and directory does not become part of the document name.

/Files=*file_name*

file_name specifies the name of the document you want to export. The *file_name* specification can include the drive and directory where the file exported is to be placed. The subdirectory specified here must already exist.

You can use the wildcard characters '*' and '?' in the name part of the parameter to export all documents matching the given name pattern (e.g. "C:\DOCEXP*.DOC" to export all documents with a file name extension of ".DOC" to the directory "C:\DOCEXP").

If you specify several documents, enclose the document names in brackets and separate them by commas.

If you want to specify a list of document names in a list file, refer to this list file by preceding the list file name with the @ symbol (@ *list_file_name*). In the list file enclose the document names in brackets and separate them by commas.

/Options=

Specifies whether you want to export an original document (SOURCE), a translation (TARGET), a document with nonmatching segments (SNOMATCH), or a validation document (VALDOC).

/Val=

Specifies additional options for the export of validation documents. These options control the format of the validation document (XML, HTML, DOC or ODT), the inclusion of protected segments (PROTSEGS) and allows to combine all exported documents into one validation document (COMBINE). If you specify more than one option, you must enclose them in brackets and separate the individual options using a comma.

/Overwrite=

Specifies whether an existing document is overwritten when exporting a document with the same name (NO is the default).

/QUIET

If you specify the parameter in this format, you are not prompted with any message window. If you do not specify any of the /quiet parameters, a message window pops up whenever an error occurs or when the task has successfully completed.

/QUIET=NOMSG

If you specify this parameter, you are not prompted with any message window. Any error message text is routed to stdout.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters.

Examples

The following example shows how to export a document using the EQFBATCH command with the TASK option.

Example

```
eqfbatch /task=docexp /fld=pharma /files=c:\med.txt /options=target  
/overwrite=yes
```

In this example, the document to be exported is called *med.txt*. It is contained in folder *pharma*. The translation of *med.txt* is exported to the root directory of drive C. If *med.txt* already exists, it is overwritten.

The following example shows how to export documents in external format with a path.

Example

```
eqfbatch /task=docexp /fld=sample1 /startpath=i:\transl  
/files=(test1\demo1.scr,test1\demo2.scr)
```

The document test1\demo1.scr is exported to i:\transl\test1\demo1.scr and the document test1\demo2.scr is exported to i:\transl\test1\demo2.scr.

The following example shows how to export all documents of the folder as validation format documents in HTML format and to combine them into a single document.

Example

```
eqfbatch /task=docexp /fld=pharma /files=c:\valdoc\*.* /options=VALDOC  
/val=(HTML, COMBINE) /overwrite=yes
```

In this example, all documents of the folder pharma are exported in validation format and are combined into one HTML document. The combined document is stored in the directory C:\valdoc\ and has a name of pharma.html (When several documents are combined into one validation document the name of the folder is used for the combined document).

Importing documents

Purpose

documentEQFBATCH: importing (from command area) importingEQFBATCH: documents (from command area) If you have many large documents that you want to import, using the command area of your operating system instead of OpenTM2 windows might be more convenient.

Before importing, ensure that:

- **OpenTM2** has been started.
- A folder has been created. This folder can still be empty and will later contain the documents to be imported.

The properties of the folder give information on the used markup language and other document-related information that is necessary when importing the respective documents.

To work from the command area, use the **EQFBATCH** command. Its syntax is described in the following chapter.

Format

EQFBATCHimporting documents commandsEQFBATCHimporting documents The following syntax diagram describes how to import from the command area of your operating system. Note that lowercase letters in a parameter name are optional and need not be typed.

**EQFBATCH /TAsk=DOCIMP /FLD=folder /SStartpath=startpath /FFiles= file_name (,file_name)
@list_file_name /ALIAS=alias /OVerwrite= NO
YES /QUIET /QUIET=NOMSG /EDit=editor /MArkup=markup /CONV=conv**

Parameters

/TAsk=DOCIMP

DOCIMP specifies that you want to import a document.

/FLD=folder

folder specifies the name of the folder where the documents imported are to be placed.

/SStartpath=startpath

startpath specifies the drive, directory, or both, to be omitted from the document's name. It only applies to documents imported in external format where the path in which the document resides normally becomes part of the document's name.

/FFiles=file_name

file_name specifies the name of the document you want to import. The file_name specification can include the drive and directory where the file to be imported is found.

If you specify several documents, enclose the document names in brackets and separate them by commas.

If you want to specify a list of document names in a list file, refer to this list file by preceding the list

file name with the @ symbol (@ *list_file_name*). In the list file enclose the document names in brackets and separate them by commas.

/ALIAS=alias

alias specifies an alias name for the document to be imported. If you import several documents, this alias name applies to all of them. The name can be up to 256 characters long.

/OVerwrite=

Specifies if an existing document will be overwritten when importing a document with the same name (NO is the default).

/QUIET

If you specify the parameter in this format, you are not prompted with any message window. If you do not specify any of the /quiet parameters, a message window pops up whenever an error occurs or when the task has successfully completed.

/QUIET=NOMSG

If you specify this parameter, you are not prompted with any message window. Any error message text is routed to stdout.

/EDit=editor

Specifies the editor to be used for the imported document.

/MArkup=markup

Specifies the markup table to be used for the document.

/CONV=conv

Specifies the conversion to be used for the document.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to import several documents using the EQFBATCH command.

Example

```
eqfbatch /task=docimp /fld=pharma /files=@medlist.lst /overwrite=no
```

In this example, the documents to be imported are listed in file medlist.lst. medlist.lst looks as follows:

medlist.lst

```
(d:\medset.doc, d:\medfrm.doc, d:\medb01.doc, d:\medb02.doc)
```

The file medlist.lst must be located either on the current directory, or (if it is not in the current directory) you must specify the complete path where it is located. For example,
@d:\test\medlist.lst.

Note:

There is a @ sign before the drive letter.

All files contained in medlist.lst are to be imported into folder pharma and are located in the root directory of drive D. If one of the files already exists in the specified folder, it is not overwritten and you get a message.

Examples

The following examples show how to import documents in external format with a path.

Assume that the following directories and files are on your V disk:

```
[proj1]
[vers1]
[mri]
base.mri
dialog.mri
dialog2.mri
message.osm
[docs]
intro.htm
lesson.htm
```

```
vers2]
[mri]
base.mri
dialog.mri
dialog2.mri
message.osm
[docs]
intro.htm
lesson.htm
```

Example 1

```
eqfbatch /task=docimp /fld=test /startpath=v:\ /files=(vers1\docs\*.htm,vers2\docs\*.htm)
```

In this example, the following documents are imported into folder test:

```
vers1\docs\intro.htm
vers1\docs\lesson.htm
vers2\docs\intro.htm
vers2\docs\lesson.htm
```

Example 2

```
eqfbatch /task=docimp /fld=test /startpath=v:\vers1\ /files=(mri\*.*,docs\*.*)
```

In this example, the following documents are imported into folder test:

```
mri\base.mri
mri\dialog.mri
mri\dialog2.mri
mri\message.osm
docs\intro.htm
docs\lesson.htm
```

Example 3

```
eqfbatch /task=docimp /fld=test /startpath=v:\vers1\ /files=(*.*)
```

In this example, the following documents are imported into folder test:mri\base.mri

```
mri\dialog.mri
mri\dialog2.mri
mri\message.osm
docs\intro.htm
docs\lesson.htm
```

Example 4

```
eqfbatch /task=docimp /fld=test /startpath=v:\ /files=(*.*)
```

In this example, the following documents are imported into folder test:

```
vers1\mri\base.mri
vers1\mri\dialog.mri
vers1\mri\dialog2.mri
vers1\mri\message.osm
vers1\docs\intro.htm
vers1\docs\lesson.htm
vers2\mri\base.mri
vers2\mri\dialog.mri
vers2\mri\dialog2.mri
vers2\mri\message.osm
vers2\docs\intro.htm
vers2\docs\lesson.htm
```

Example 5

```
eqfbatch /task=docimp /fld=test /startpath=v:\vers1\docs /files=(*.htm)
```

In this example, the following documents are imported into folder test:

```
intro.htm
lesson.htm
```

Creating a folder

Purpose

EQFBATCH: creating (from command area) folders (from command area) If you need to set up new folders frequently, using the command area of your operating system instead of OpenTM2 windows might be more convenient. To work from the command area, use the **EQFBATCH** command. Its syntax is described in the following chapter.

Format

EQFBATCHcreating folders commandsEQFBATCHcreating folders The following syntax diagram describes how you create a folder from the Windows^(R) command area. Note that lowercase letters in a parameter name are optional and need not be typed.

**EQFBATCH /TAsk=FLDCRT /NAme=folder
/DEsc=desc /TOdrive=drive /MEm=memdb /MArku=markuptable /EDit=editor /DIct= dict (,dict)
/SRcIn=source /TGtIn=target /QUIET /QUIET=NOMSG**

Parameters

/TAsk=FLDCRT

FLDCRT specifies that you want to set up a new folder.

/NAme=folder

folder specifies the name of the new folder.

/DEsc=desc

desc specifies the description of the new folder (up to 40 characters). If none is specified, the folder will have no description and will be more difficult to relate to a specific project later on.

/TOdrive=drive

drive specifies the drive where the new folder is to be placed. It must be either the main drive or a drive that has been configured as additional drive.

/MEm=memdb

memdb specifies the name of an existing Translation Memory that is to be related to the new folder.

/MArku=markuptable

markuptable specifies the name of a markup table that is to be related to the markup language of the documents that will be stored in the new folder.

/EDit=editor

editor specifies the name of an editor that is to be associated with the new folder (only STANDARD is possible).

/DIct=dict

dict specifies the name of a dictionary to be associated with the new folder. This parameter is optional, that is, it is not required that a dictionary is specified. If you specify several dictionaries, enclose the dictionary names in brackets and separate them by commas.

/SRcIn=source

source is the source language of the documents stored in the new folder.

/TGtIn=target

target is the target language of the documents stored in the new folder.

/QUIET

If you specify the parameter in this format, you are not prompted with any message window. If you do not specify any of the /quiet parameters, a message window pops up whenever an error occurs or when the task has successfully completed.

/QUIET=NOMSG

If you specify this parameter, you are not prompted with any message window. Any error message text is routed to stdout.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to create a folder using the EQFBATCH command.

Example

```
eqfbatch /task=fldcrt /name=proj1 /desc="new folder" /todrive=e  
/mem=ibmmem /markup=eqfbook /edit=standard /dict=(ibmterm,ibm2)  
/srcLng=German(national) /tgtLng=English(U.S.)
```

In this example, the folder to be created is called `proj1` and is described as new folder. It will be located on drive E and will have the following translation resources attached to it:

- The Translation Memory `ibmmem`
- The dictionaries `ibmterm` and `ibm2`
- The markup table `eqfbook`
- The editor `standard`

The source language of the folder's documents will be German (national), the target language will be English (U.S.).

Deleting a folder

Purpose

EQFBATCH: deleting folders If you want to delete a folder, using a command typed in your system's command area instead of OpenTM2 windows might be quicker and more convenient.

Format

EQFBATCH: deleting folders The following syntax diagram describes how you delete a folder with the EQFBATCH command. Note that lowercase letters in a parameter name are optional and need not be typed.

EQFBATCH /TAsk=FLDDEL /FLD=folder /QUIET /QUIET=NOMSG

Parameters

/TAsk=FLDDEL

`FLDDEL` specifies that you want to remove a folder.

/FLD=folder

Specifies the name of the folder that you want to remove.

/QUIET

If you specify the parameter in this format, you are not prompted with any message window. If you do not specify any of the `/quiet` parameters, a message window pops up whenever an error occurs or when the task has successfully completed.

/QUIET=NOMSG

If you specify this parameter, you are not prompted with any message window. Any error message text is routed to `stdout`.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to delete a folder using the EQFBATCH command.

Example

```
eqfbatch /task=foldel /fld=pharma /quiet
```

In this example, the folder `pharma` is deleted without prompting the user.

Exporting a folder

Purpose

EQFBATCH: exporting If you need to set up new folders frequently, using the command area of your operating

system instead of OpenTM2 windows might be more convenient. To work from the command area, use the **EQFBATCH** command. Its syntax is described in the following chapter.

Format

EQFBATCH exporting folders commands **EQFBATCH** exporting folders The following syntax diagram describes how you export a folder from the Windows^(R) command area. Note that lowercase letters in a parameter name are optional and need not be typed.

```
EQFBATCH /TAsk=FLDEXP /FLD=folder /FIlles= document_name (,document_name )
@list_file_name /TOdrive=drive /ToPath=\EQF\EXPORT /ToPath=path /OPtions= (,
)/OVerwrite= YES NO /DEsc= desc @desc_file_name /QUIET /QUIET=NOMSG option DICT
MEM ROMEM DOCMEM DELETE XLIFF MASTERFOLDER
```

Parameters

/TAsk=FLDEXP

FLDEXP specifies that you want to export a folder.

/FLD=folder

folder specifies the name of the folder that you want to export.

/FIlles=document_name

document_name specifies the name of a document to be exported.

If you specify several documents, enclose the document names in brackets and separate them by commas.

If you want to specify a list of document names in a list file, refer to this list file by preceding the list file name with the @ symbol (@ *list_file_name*). In the list file enclose the document names in brackets and separate them by commas.

If you do not specify the **FIlles** option, all documents in the folder are exported.

/TOdrive=drive

drive specifies the drive where the folder is to be exported to.

/ToPath=path

path specifies the path where the folder is to be exported to. Start the path value with a backslash \. The default is \eqf\export.

/OPtions=

Specifies whether you want to export the folder with its dictionaries (DICT), with its Translation Memory (MEM), with its Translation Memory databases to be searched (ROMEM), with its **Translation Memory** and all the **Translation Memory databases** defined for its documents (DOCMEM), or whether you want to delete the folder after it has been exported (DELETE), or whether you want to export the folder as a master folder (MASTERFOLDER) or as a XLIFF translation package (XLIFF).

A master folder exported without the MASTERFOLDER option becomes a child folder when it is re-imported. If you specify more than one option, you must enclose them in brackets. The XLIFF option cannot be used together with the options DELETE and MASTERFOLDER.

/OVerwrite=

Specifies whether an existing folder is overwritten when exporting a folder with the same name (NO is the default).

/DEsc=desc

desc specifies the description that you want to add to the exported folder, or the name of a file containing the description. The text that gives more information about the folder must be enclosed in quotes. If you specify a file that contains the text, the name of the file must be preceded by the @ symbol.

/QUIET

If you specify the parameter in this format, you are not prompted with any message window. If you do not specify any of the /quiet parameters, a message window pops up whenever an error occurs or when the task has successfully completed.

/QUIET=NOMSG

If you specify this parameter, you are not prompted with any message window. Any error message text is routed to stdout.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to export a folder using the EQFBATCH command.

Example

```
eqfbatch /task=fldexp /fld=pharma /files=(161.scr,1500.scr) /todrive=e  
/options=(dict,mem) /desc="This is information on the folder"
```

In this example, the folder to be exported is called `pharma`. It is exported to drive E with the documents `161.scr` and `1500.scr`, with its associated dictionary and Translation Memory. The note `This is information on the folder` is added to the folder.

Importing a folder

Purpose

folderEQFBATCH: importing (from command area) **importingEQFBATCH:** folders (from command area) It might be more convenient to use a command instead of OpenTM2 windows to import a folder. This is especially useful if you have many folders that you want to import.

Format

EQFBATCHimporting folders commandsEQFBATCHimporting folders The following syntax diagram describes how you import a folder from the Windows^(R) command line. Note that lowercase letters in a parameter name are optional and need not be typed.

```
EQFBATCH  
/TAsk=FLDIMP  
/FLD=folder /FRomdrive=drive /FromPath=\EQF\EXPORT /FromPath=path /OPtions= (,  
)  
/TOdrive=drive  
/ToPath=\EQF\EXPORT  
/ToPath=path /QUIET /QUIET=NOMSG /EDit=editor /MArkuP=markup /CONV=conv option DICT  
XLIFF MEM
```

Parameters**/TAsk=FLDIMP**

`FLDIMP` specifies that you want to import a folder.

/FLD=folder

`folder` specifies the name of the folder that you want to import.

/FRomdrive=drive

`drive` specifies the drive where the folder is imported from.

/FromPath=path

`path` specifies the path where the folder is imported from. Start the path value with a backslash \. The default is `\eqf\export`.

/OPtions=

Specifies whether you want to import the folder with its associated dictionary (`DICT`), with its Translation Memory (`MEM`) or if you want to import a XLIFF package (`XLIFF`). If you specify more than one option, you must enclose them in brackets and separate them with commas.

/TOdrive=drive

`drive` specifies the drive where the folder is to be imported to. If nothing is specified, the folder is imported to the drive where the OpenTM2 program files are located.

/ToPath=path

`path` specifies the path where the folder is to be imported to. Start the path value with a backslash \. The default is `\eqf\export`.

/QUIET

If you specify the parameter in this format, you are not prompted with any message window. If you do not specify any of the /quiet parameters, a message window pops up whenever an error occurs or when the task has successfully completed.

/QUIET=NOMSG

If you specify this parameter, you are not prompted with any message window. Any error message text is routed to stdout.

/EDit=editor

Specifies the editor to be used for the imported folder.

/MArket=markup

Specifies the markup table to be used for the imported folder.

/CONV=conv

Specifies the conversion to be used for the imported folder.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to import a folder using the EQFBATCH command.

Example

```
eqfbatch /task=fldimp /fld=pharma /fromdrive=a /todrive=e /options=(dict,mem)
```

In this example, the folder to be imported is called `pharma`. It is imported from drive A to drive E together with its associated dictionary and Translation Memory.

Creating a Translation Memory

Purpose

Translation MemoryEQFBATCH: creating (from command area) creatingEQFBATCH: Translation Memory databases (from command area) To set up a new Translation Memory, it might be more convenient to use a command instead of using OpenTM2 windows.

Format

EQFBATCHcreating Translation Memory databases commandsEQFBATCHcreating Translation Memory databases The following syntax diagram describes how you create a Translation Memory from the Windows^(R) command line. Note that lowercase letters in a parameter name are optional and need not be typed.

**EQFBATCH /TAsk=MEMCRT /NAme=memdb /DEsc=desc /TYpe= SHARED
LOCAL /TOdrive=drive /SRcIng=source /QUIET /QUIET=NOMSG**

Parameters

/TAsk=MEMCRT

`MEMCRT` specifies that you want to set up a new Translation Memory.

/NAme=memdb

`memdb` specifies the name of the new Translation Memory.

/DEsc=desc

`desc` specifies the description of the new Translation Memory (up to 40 characters). If this parameter is not specified, the new Translation Memory will have no description.

/TYpe=

Specifies whether the new Translation Memory is to be shared among several OpenTM2 users (SHARED) or if it is a local one which can be used only by you (LOCAL). If you omit this parameter, LOCAL is assumed.

/TOdrive=drive

`drive` specifies the drive where the new Translation Memory is to be located.

/SRcIng=source

`source` is the source language of the text segments stored in the new Translation Memory.

/QUIET

If you specify the parameter in this format, you are not prompted with any message window. If you

do not specify any of the /quiet parameters, a message window pops up whenever an error occurs or when the task has successfully completed.

/QUIET=NOMSG

If you specify this parameter, you are not prompted with any message window. Any error message text is routed to stdout.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to create a Translation Memory using the EQFBATCH command.

Example

```
eqfbatch /task=memcrt /name=medmem /desc="new memory" /type=local  
/todrive=e /srclng=English(U.S.)
```

In this example, the new Translation Memory `medmem` is a local one and will be located on drive E. The source language of the stored text segments is English (U.S.).

Deleting a Translation Memory

Purpose

EQFBATCH: deleting Translation Memory databases (from command area) **EQFBATCH**: Translation Memory databases (from command area) If you want to delete a Translation Memory, using a command typed in your system's command area instead of OpenTM2 windows might be quicker and more convenient.

Format

EQFBATCH: deleting Translation Memory databases commands **EQFBATCH**: Deleting Translation Memory databases The following syntax diagram describes how you delete a Translation Memory with the EQFBATCH command. Note that lowercase letters in a parameter name are optional and need not be typed.

EQFBATCH /TAsk=MEMDEL /MEm=memdb /QUIET /QUIET=NOMSG

Parameters

/TAsk=MEMDEL

`MEMDEL` specifies that you want to remove a Translation Memory.

/MEm=memdb

Specifies the name of the Translation Memory that you want to remove.

/QUIET

If you specify the parameter in this format, you are not prompted with any message window. If you do not specify any of the /quiet parameters, a message window pops up whenever an error occurs or when the task has successfully completed.

/QUIET=NOMSG

If you specify this parameter, you are not prompted with any message window. Any error message text is routed to stdout.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to delete a Translation Memory using the EQFBATCH command.

Example

```
eqfbatch /task=memdel /mem=medmem
```

In this example, the Translation Memory `medmem` is deleted. You will be prompted with a message window.

Exporting a Translation Memory

Purpose

Translation MemoryEQFBATCH: exporting (from command area) exportingEQFBATCH:
Translation Memory databases (from command area) If you want to export a Translation Memory, using a command typed in your system's command area instead of OpenTM2 windows might be more convenient.

Format

EQFBATCHexporting Translation Memory databases commandsEQFBATCHexporting Translation Memory databases The following syntax diagram describes how you export a Translation Memory from the Windows^(R) command line. Note that lowercase letters in a parameter name are optional and need not be typed.

**EQFBATCH /Task=MEMEXP /OUT=mem_file /TYpe= ASCII ANSI UTF16 TMXUTF8
TMXUTF16 /MEm=memdb /OVerwrite= NO YES /QUIET /QUIET=NOMSG**

Parameters

/Task=MEMEXP

MEMEXP specifies that you want to export a Translation Memory.

/OUT=mem_file

mem_file specifies the name (along with the drive and directory) of a file where the Translation Memory specified in memdb is to be exported to. The file must have the extension EXP.

/TYpe=

- ASCII specifies that the encoding of the exported memory should be in ASCII and the format is SGML (EXP).
- ANSI specifies that the encoding of the exported memory should be in ANSI and the format is SGML (EXP).
- UTF16 specifies that the encoding of the exported memory should be in Unicode (UTF-16) and the format is SGML (EXP).
- TMXUTF8 specifies that the encoding of the exported memory should be in UTF-8 and the format is TMX.
- TMXUTF16 specifies that the encoding of the exported memory should be in UTF-16 and the format is TMX.

/MEm=memdb

memdb specifies the name of the Translation Memory that you want to export

/OVerwrite=

Specifies whether an existing Translation Memory file is to be overwritten when exporting the Translation Memory.

/QUIET

If you specify the parameter in this format, you are not prompted with any message window. If you do not specify any of the /quiet parameters, a message window pops up whenever an error occurs or when the task has successfully completed.

/QUIET=NOMSG

If you specify this parameter, you are not prompted with any message window. Any error message text is routed to stdout.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to export a Translation Memory using the EQFBATCH command.

Example

```
eqfbatch /task=memexp /out=d:\mymemo.exp /type=external /mem=oldmemo
```

In this example, the Translation Memory `oldmemo` is exported to the file `d:\mymemo.exp`.

Importing a Translation Memory

Purpose

Translation MemoryEQFBATCH: importing (from command area) importingEQFBATCH:
Translation Memory databases (from command area) To import a Translation Memory, you might find it more convenient to use a command instead of OpenTM2 windows. You might want to do this if you have many Translation Memory databases that you want to import.

Format

EQFBATCHImporting Translation Memory databases commandsEQFBATCHImporting Translation Memory databases The following syntax diagram describes how you import a Translation Memory from the command area. Note that lowercase letters in a parameter name are optional and need not be typed.

**EQFBATCH /TAsk=MEMIMP /FIlles=mem_file /TYpe= ASCII ANSI UTF16 TMX
TMXTRADOS /MEM=memdb /QUIET /QUIET=NOMSG**

Parameters

/TAsk=MEMIMP

`MEMIMP` specifies that you want to import a Translation Memory.

/FIlles=mem_file

`mem_file` specifies the name of the Translation Memory that you want to import along with the drive and directory where it is located.

/TYpe=

- `ASCII`
specifies that it is an external Translation Memory (in SGML format) encoded in ASCII that you want to import.
- `ANSI`
specifies that it is an external Translation Memory (in SGML format) encoded in ANSI that you want to import.
- `UTF16`
specifies that it is an external Translation Memory (in SGML format) encoded in Unicode (UTF-16) that you want to import.
- `TMX`
specifies that it is an external TMX Translation Memory that you want to import (the encoding can be either in UTF-8 or UTF-16).
- `TMXTRADOS`
specifies that it is an external TMX Translation Memory that you want to import (the encoding can be either in UTF-8 or UTF-16) and what RTF tags should be removed from the imported data.

/MEM=memdb

`memdb` specifies the name of an existing Translation Memory to which the Translation Memory specified in `mem_file` is to be imported to. The contents of these Translation Memory databases will then be merged.

/QUIET

If you specify the parameter in this format, you are not prompted with any message window. If you do not specify any of the /quiet parameters, a message window pops up whenever an error occurs or when the task has successfully completed.

/QUIET=NOMSG

If you specify this parameter, you are not prompted with any message window. Any error message text is routed to stdout.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to import a Translation Memory using the EQFBATCH command.

Example

```
eqfbatch /task=memimp /files=d:\mymemo.mem /type=UTF16 /mem=oldmemo
```

In this example, the Translation Memory `mymemo.mem`, which is in external format, is imported to the Translation Memory `oldmemo`.

Opening a document

Purpose

organizingEQFBATCH: Opening a document Translation MemoryEQFBATCH: Opening a document Instead of using OpenTM2 windows to open a document, you might find it more convenient to use a command.

Format

EQFBATCHorganizing **Translation Memory databases** commandsEQFBATCHorganizing Translation Memory databases The following syntax diagram describes how you open a document from the Windows® command line. Note that lowercase letters in a parameter name are optional and need not be typed.

EQFBATCH
/TAsk=DOCOPEN
/FLD=folder /NAme=document_name /SEgment=segnumber /LIne=linenumber /QUIET

Parameters

/TAsk=DOCOPEN

`DOCOPEN`specifies that you want to open a document.

/FLD=folder

`folder` specifies the name of the folder that contains the document you want to open.

/NAme=document_name

`document_name` specifies the name of the document to be opened.

/SEgment=segnumber

`segnumber` specifies the number of the segment which is to be activated when the document is opened.

/LIne=linenumber

`linenumber` specifies the line number to which the cursor is positioned when the document is opened.

/QUIET

If you specify this parameter, you are not prompted with any message window. If you do not specify this parameter, a message window pops up whenever an error occurs or when the task has successfully completed.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to organize a **Translation Memory** using the EQFBATCH command.

Example

```
eqfbatch /task=docopen /fld=sample1 /name=device.scr
```

This command opens the document DEVICE.SCR which is stored in folder SAMPLE1.

Organizing a Translation Memory

Purpose

organizingEQFBATCH: **Translation Memory databases** (from command area) Translation MemoryEQFBATCH: organizing (from command area) To organize a **Translation Memory** it might be more convenient to use a command instead of **OpenTM2** windows. It is particularly useful if you want to regularly organize several or all **Translation Memory databases** overnight.

Format

EQFBATCHorganizing Translation Memory databases commandsEQFBATCHorganizing Translation Memory databases The following syntax diagram describes how you organize a Translation Memory with the EQFBATCH command. Note that lowercase letters in a parameter name are optional and need not be typed.

EQFBATCH /TAsk=MEMORG /MEm= memdb (,memdb) @list_member_name

Parameters

/TAsk=MEMORG

MEMORG specifies that you want to organize a **Translation Memory**.

/MEm=memdb

memdb specifies the name of the Translation Memory to be organized.

If you want to specify several **Translation Memory databases**, enclose the Translation Memory database names in brackets and separate them by commas.

If you want to specify a list of Translation Memory database names in a list file, refer to this list file by preceding the list file name with the @ symbol (@ list_member_name). In the list file enclose the Translation Memory database names in brackets and separate them by commas.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to organize a **Translation Memory** using the EQFBATCH command.

Example

eqfbatch /task=memorg /mem=sample1

In this example, the Translation Memory sample1 is organized.

Renaming a folder, a dictionary or a Translation Memory

Purpose

organizingEQFBATCH: Renaming a folder, a dictionary or a Translation Memory (from command area) Translation MemoryEQFBATCH: rename (from command area) If you want to rename a folder, a dictionary or a Translation Memory using a command typed in your system's command area instead of OpenTM2 windows might be quicker and more convenient.

Format

EQFBATCHrename a folder, a dictionary or a Translation Memory commandsEQFBATCHrename a folder, a dictionary or a Translation Memory The following syntax diagram describes how you rename a folder, a dictionary or a TranslationMemory with the EQFBATCH command. Note that lowercase letters in a parameter name are optional and need not be typed.

For the rename of folders:

EQFBATCH /TAsk=RENAME /FLD=folder /NEW=newname /QUIET /QUIET=NOMSG

For the rename of dictionaries:

EQFBATCH /TAsk=RENAME /Dlct=dicname /NEW=newname /QUIET /QUIET=NOMSG

For the rename of Translation Memory databases:

**EQFBATCH
/TAsk=RENAME /MEm=memdb /NEW=newname /ADJust=YES /QUIET /QUIET=NOMSG**

Parameters

/TAsk=RENAME

RENAME specifies that you want to rename a folder, dictionary or a TranslationMemory.

/FLD=folder

Specifies the name of the folder that you want to rename.

/DICT=dicname

Specifies the name of the dictionary that you want to rename.

/MEM=memdb

Specifies the name of the Translation Memory that you want to rename.

/NEW=newname

Specifies the new name for the folder, dictionary or Translation Memory.

/ADJust=YES

When specified EQFBATCH will change all occurrences of the TranslationMemory name or dictionary name in the properties of folders and documents

/QUIET

If you specify the parameter in this format, you are not prompted with any message window. If you do not specify any of the /quiet parameters, a message window pops up whenever an error occurs or when the task has successfully completed.

QUIET=NOMSG

If you specify this parameter, you are not prompted with any message window. Any error message text is routed to stdout.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to rename a folder using the EQFBATCH command.

Example

```
eqfbatch /task=rename /fld=pharma /new=pharmaceutical /quiet
```

In this example, the folder pharma is renamed to pharmaceutical without prompting the user.

The following example shows how to rename a TranslationMemory and adjust all references to the Translation Memory name automatically.

Example

```
eqfbatch /task=rename /mem=MyMem /new=MyNewMem /adjust=yes
```

In this example, the Translation Memory MyMem is renamed to MyNewMem and all references to this memory in the properties of folders and documents are adjusted to use the new name.

Performing several tasks in one step

Purpose

Instead of typing all parameters in one line, you can use the /TASKLIST option and specify the parameters for any task in a separate file.

Format

EQFBATCH /TASKLIST=list_of_tasks /QUIET /QUIET=NOMSG

Parameters**/TASKLIST=list_of_tasks**

list_of_tasks is the name of a file containing a task description on each line, for example, parameters for several import or export requests (see examples).

/QUIET

If you specify the parameter in this format, you are not prompted with any message window. If you do not specify any of the /quiet parameters, a message window pops up whenever an error occurs or when the task has successfully completed.

/QUIET=NOMSG

If you specify this parameter, you are not prompted with any message window. Any error message text is routed to stdout.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to specify the export parameters in a separate file (taskexp.fil) to export several documents (med1.txt and med2.txt) and then call the EQFBATCH command with a reference to this file.

taskexp.fil

```
/task=docexp /fld=pharma /files=c:\med1.txt /options=target  
/overwrite=yes  
/task=docexp /fld=pharma /files=c:\med2.txt /options=target /overwrite=yes
```

Enter the EQFBATCH command as follows:

Example

```
eqfbatch /tasklist=taskexp.fil
```

This lets you export more than one document with one command. All EQFBATCH tasks can be combined.

The next example shows how to specify the import parameters in a separate file (taskimp.fil) to import several documents listed in two files (@meddoca.lst and @meddocb.lst) and then call the EQFBATCH command with a reference to this file.

taskimp.fil

```
/task=docimp /fld=pharma /files=@meddoca.lst /overwrite=no  
/task=docimp /fld=pharma /files=@meddocb.lst /overwrite=no
```

Enter the EQFBATCH command as follows:

Example

```
eqfbatch /tasklist=taskimp.fil
```

This lets you import more than one document with one command. All EQFBATCH tasks can be combined.

Working from the command area (Other Tools)

This chapter contains other tools which can be executed from command area.

To start a task from the command area:

1. Type a command and its parameters in your operating system's command area (next to the command prompt) according to the command syntax described in the following chapters.
2. Press Enter.

Note that lowercase letters in a parameter name are optional and need not be typed. A mixture between uppercase and lowercase letters is allowed. However, a muddled letter sequence like in OVerWRiTTe must not be used. The parameters can be entered in any order whatever.

To process several commands in sequence, type the individual commands into a batch file (*.BAT or *.CMD) using any text editor. You can execute these commands by starting the batch file.

To avoid retyping of long text strings in commands, you can also type a command and its parameters in a text editor, which is capable of the **Copy to clipboard** function. You can then **paste** the text string from the clipboard directly into your system's command area and start the task.

How to read syntax diagrams

In this chapter diagrams are used to illustrate the programming syntax. To use a diagram, follow a path from left to right, top to bottom, adding elements as you go. In these diagrams, all spaces and other characters are significant.

Each diagram begins with a double right arrowhead and ends with a right and left arrowhead pair. Lines beginning with single right arrowheads are continuation lines.

keyword=variable_value

Keywords are all in lowercase, but can be entered in uppercase or in lowercase. Variable values that you provide are shown in *italics* and are usually in lowercase. Where values are shown in uppercase, they should be entered as they appear.

In a choice of items, the default item is always shown above the main line:

keyword= default_value other_value other_value

Optional syntax elements are shown below the main line:

keyword=value

A repeat arrow shown above an item or a stack of items indicates that you can specify the item multiple times or specify more than one of the items. A character (such as a comma) on the repeat arrow line indicates that the items must be separated by that character. A repeatable operand is shown like this:

keyword= (,variable_value)

Syntax diagrams can be broken into fragments. A fragment is indicated by vertical bars with the name of the fragment between the bars. The fragment is shown following the main diagram, like so:

/KEYWORD= ITEM1 variable1 variable2 variable3

Converting a NLV memory to a source/source memory

Purpose

For the conversion of NLV Translation Memories (containing translated segments) to source/source Translation Memories OpenTM2 offers the command *EwfCreateITMFromMemory*. This tool reads a NLV memory and writes the segments of this memory to a new memory replacing the target of the segments with the segment source.

Format

EwfCreateITMFromMemory /INmem=inmemdb /OUTmem=outmemdb

Parameters

/INmem = inmemdb

inmemdb specifies the name of the NLV Translation Memory that you want to use as input Translation Memory.

/OUTmem = outmemdb

outmemdb is the name of a new Translation Memory which is to receive the converted segments from the input Translation Memory. The Translation Memory *outmemdb* is created by *EwfCreateITMFromMemory*.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to use the *EwfCreateITMFromMemory* command. Enter the *EwfCreateITMFromMemory* command as follows:

Example

```
EwfCreateITMFromMemory /inmem=nlv-mem /outmem=source-source-mem
```

In this example all segments contained in Translation Memory *nlv-mem* are written to the new output Translation Memory *source-source-mem* and the target of the segments is replaced with the source of the segments.

Creating an Initial Translation Memory from the command line

Purpose

Initial Translation Memory (ITM) creating (from command area) creating Initial Translation Memory (ITM) databases (from command area) If you need to create Initial Translation Memory (ITM) databases frequently, you might find it more convenient to use a command instead of the procedure described in [Creating an Initial Translation Memory](#).

The correctness of the aligned segment pairs in an internal ITM can be checked with a text editor as described in [The Initial Translation Memory editor](#) or [Revising an Initial Translation Memory](#).

When you consider the Initial **Translation Memory** to be correct, you can begin using it for your translations.

The EQFITM command has additional options compared to the window version:

- You can create not only an internal **Translation Memory**, but also an external **Translation Memory**.
- You must not fill the internal **Translation Memory**.
- You can suppress the confirmation message.

Format

EQFITM creating an Initial Translation Memory commands **EQFITM** creating an Initial Translation Memory
The following syntax diagrams describe how you create an initial **Translation Memory** from the Windows^(R) command area. Note that lowercase letters in a parameter name are optional and need not be typed.

EQFITM /MEm=memdb³ /Ffiles= @list_file_name —
⁴
/MArket=markup
/SGMLmem=mem_file
/SRCIng=source_language
/TGtIng=target_language
/SrcStartpath=source_startpath /TgtStartpath/TS=target_startpath /TYpe=(,)
/sglformat=format UNICODE ASCII ANSI file_pairs(,file_name_original,file_name_translation)
type NOANA NOTM NOCONF PREPARE VISUAL

Parameters

/SGMLFORMAT=format

format specifies the export format of a ITM. The default parameter is UNICODE. Other possible parameters are ANSI and ASCII. Shortform is /sf.

/MEm=memdb

memdb is the name of a previously created **OpenTM2 Translation Memory** (without the file extension). This **Translation Memory** can still be empty. It can be filled with original segments and their corresponding translations.

/Ffiles=file_pairs

file_pairs specifies the pair of files to use when creating the ITM. Enclose the pair of files in brackets and separate the file names by a comma.

If you specify several pairs (as original1, translation1, original2, translation2, and so on), enclose the file names in brackets and separate them by commas.

If you want to specify a list of pairs in a list file, refer to this list file by preceding the list file name with the @ symbol (@ *list_file_name*). In the list file enclose the file names in brackets and separate them by commas.

/MArket=markup

markup is the name of the markup table that the selected texts use (mandatory except for /TYPE=NOANA). See [Working with markup tables](#) for the names of the markup tables.

/SGMLmem=mem_file

mem_file is the name you want to give to the external ITM and the path where it is to be located. The ITM is in SGML format and can subsequently be imported into **OpenTM2** after you have checked it.

/SRCIng=source_language

source_language is the language of the original documents.

/TGtIng=target_language

target_language is the language of the translated documents.

/SrcStartpath=source_startpath

source_startpath is the path information that you do **not** want to become part of the document name when the original document is stored in the **Initial Translation Memory**. For example, if your source file is stored in e:\tm\project\english and you do not want e:\tm\project to

3 Mandatory, but not filled if /TYPE=NOTM

4 Mandatory except for /TYPE=NOANA

become part of the name under which it is stored, specify `e:\tm\project` in this field.

The path you specify here can differ from the `target_startpath`. However, if you specify a source start path, you must also specify a `target_startpath`.

/TgtStartpath=target_startpath

`target_startpath` is the path information that you do **not** want to become part of the document name when the target document is stored in the **Initial Translation Memory**.

For example, if your source file is stored in `e:\tm\project\english` and you do not want `e:\tm\project` to become part of the name under which it is stored, specify `e:\tm\project` in this field. The path you specify here can differ from the `source_startpath`. However, if you specify a source start path, you must also specify a `source_startpath`.

/TYpe=

One or more of the following:

- NOANA
Do not analyze the selected files because they have already been analyzed by **OpenTM2**.
- NOTM
Do not fill the internal **Translation Memory** (`memdb`). Fill the external **Translation Memory**. It is in SGML format and you can check it afterwards.
- NOCONF
No confirmation message is displayed. This is useful if you do not want to be interrupted by progress messages while you are working.
- PREPARE
The source documents are related to their corresponding translations. The file pairs are prefixed with **p**.
- VISUAL Creates the ITM and presents its contents on the screen for you to revise.

Examples

The following examples show how to use the EQFITM command.

Example:

```
eqfitm /mem=wpitm /files=(d:\eng\text.doc,d:\ger\text.doc)
        /markup=eqfwp /srcIngr=English(U.S.) /tgtIngr=German(national)
```

Note:

For better readability, the command is shown on two lines here. However, it must be typed in one line.

In this example, the original WordPerfect document `text.doc` is located in the directory `d:\eng\`. The German translation is located in the directory `d:\ger\`. `eqfwp` is the name of the **OpenTM2** markup table used for this document. All matching sentence pairs are put into **Translation Memory** `wpitm`. No external **Translation Memory** (SGML format) is created.

Example:

```
eqfitm /mem=myitm /files=@itm.lst /markup=eqfami /sgmlmem=xx.out
        /type=notm /srcIngr=English(U.S.) /tgtIngr=German(national)
```

Note:

For better readability, the command is shown on two lines here. However, it must be typed in one line.

In this example, a **Translation Memory** `myitm` is used to find out the source language. The files to be analyzed are contained in `itm.lst`. `itm.lst` looks as follows:

`itm.lst`

```
(  
d:\eng\text1.doc, d:\ger\text1.doc,  
d:\eng\text2.doc, d:\ger\text2.doc  
)
```

The ITM created is placed into an SGML **Translation Memory** named `xx.out`, not into `myitm` (because type `notm` is specified). The markup table to be used is `eqfami` because the documents were originally created with Ami Pro.

You must type the command in one line. You can type it in uppercase letters, lowercase letters, or mixed case. You cannot use any wildcard characters, such as asterisks (*).

Note:

It is recommended to proofread the external ITM before using it in **OpenTM2**. Because it has not been created during translation, but consists of machine-generated matches, the alignment of original and translated segments might not be completely correct. This could be the case where two original segments have been combined into one translated segment.

Revising an Initial Translation Memory

Purpose

Initial Translation Memory (ITM) checking segment pairs Translation Memory matches machine-generated matches machine-generated Initial Translation Memory (ITM) checking correctness of matches All matches placed in an Initial Translation Memory are indicated by an [m] prefix. It is recommended that you check the correctness of these matches before using them if you have not already done this using the Initial Translation Memory editor.

An alternative method to perform this check is to retranslate the original document now with OpenTM2, and to compare the sentences in the original document with the translation proposals from the Initial Translation Memory. For each sentence, you can either accept or reject the saved translation. If you accept the translation proposal, the [m] flag is removed. Translations you reject, however, retain the [m] flag. Finally, the EQFDMM command deletes all translations that still have the machine translation flag.

To check the correctness of a newly generated Initial Translation Memory, proceed as follows:

1. Create a folder with the following properties:
 - The name of the Initial Translation Memory as **Translation Memory** name.
 - The same markup language that was used when creating the Initial Translation Memory.
 - The same source and target language you used when creating the Initial Translation Memory.
 - If you want to make changes while checking, select the appropriate dictionaries.
2. Import only the original documents into the folder, not the target documents. Use the same original files that you used when creating the Initial Translation Memory.
3. Open each document and perform the following tasks for all sentences:
 - If there is an acceptable proposal for the current segment, displayed in the "Translation Memory" window, accept it by pressing Ctrl+n, where n is the number of the proposal you consider correctly translated. The source segment is replaced with the accepted target segment. Press Ctrl+Enter to store the translation of the segment now as a human translation that removes the [m] flagging and proceeds to the next segment.
 - If you do not accept any of the displayed proposals, move the cursor to the text following the currently active segment (for example, by pressing Ctrl+End and moving the cursor to the right) and then press Ctrl+Enter.
4. When you have reached the end of all documents, leave the Translation Environment.
5. When you have checked the Initial Translation Memory based on all the documents that were used to create it, the Initial Translation Memory contains exact matches (the approved proposals) and machine-generated matches (the rejected proposals).

Format

commandsEQFDMM EQFDMM The following syntax diagram describes how you remove all the rejected machine-generated matches from the Initial Translation Memory from the Windows^(R) command area. Note that lowercase letters in a parameter name are optional and need not be typed.

EQFDMM /MEm=memdb /TYpe=NOCONF

Parameters

/MEm=memdb

memdb is the name of the previously created and checked Initial Translation Memory (without the file extension). All segments with the [m] flag are removed from this **Translation Memory**, all others remain unchanged.

/TYpe=NOCONF

NOCONF specifies that no confirmation message is displayed at the end of the process. This is useful when you do not want to be interrupted by a message.

When all machine-generated matches are removed from the **Translation Memory**, this is indicated by a completion message.

Examples

The following example shows how to remove all machine-generated matches from a **Translation Memory** called *INTERITM*.

Example

`eqfdmm /mem=interitm`

Reversing a Translation Memory

Purpose

Translation Memory reversing OpenTM2 provides a command to reverse a Translation Memory. The reversed Translation Memory is saved in a new Translation Memory file. This means that the source language of the Translation Memory you want to reverse is turned into the target language. The target language of the reversed Translation Memory is now becoming the source language. This can be useful when you are working in a company which does translations from any to any languages. Or, when a document was initially written in English, translated into German, revised in German for a second release, and now must be translated back into English.

Format

EQFREVM commands EQFREVM The following syntax diagram describes how you reverse a Translation Memory from your system's command area. Note that lowercase letters in a parameter name are optional and need not be typed.

EQFREVM /MEm=memdb /REv=revdb /TYPE=NOCONF

Parameters

/MEm=memdb

memdb specifies the name of the Translation Memory that you want to reverse.

/REv=revdb

Specifies the name of the Translation Memory where you want to place the reversed Translation Memory specified in *memdb*.

/TYpe=NOCONF

No confirmation message is displayed. This is useful if you do not want to be interrupted by progress messages while you are working.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to reverse a Translation Memory using the EQFREVM command.

Example

```
eqfrevm /mem=archive /rev=std1
```

In this example, the Translation Memory `archive` is reversed into the existing Translation Memory `std1`.

Changing m-flagged segments

Purpose

Translation Memory changing m-flagged segments (from command area) Segments that were translated by machine are prefixed with an [m]. OpenTM2 provides a command to have these m prefixes removed from machine-translated segments in a Translation Memory. Alternatively, this command lets you add m flags to segments that did not have such a flag before.

Format

EQFCMM commands EQFCMM The following syntax diagram describes how you change the m flags in a Translation Memory with the EQFCMM command. Note that lowercase letters in a parameter name are optional and need not be typed.

EQFCMM /MEm=memdb /SEt= NO YES /CLear= NO YES /TYpe=NOCONF

Parameters

/MEm=memdb

memdb specifies the name of the Translation Memory that you want to work with.

/CLear=

Specifies whether you want to remove the m flags in the specified Translation Memory. This option is only required if you do not specify the SET option.

/SET=

Specifies whether you want to add new m flags to segments in the Translation Memory specified. This option is only required if you do not specify the CLEAR option.

/TYpe=NOCONF

No confirmation message is displayed. This is useful if you do not want to be interrupted by progress messages while you are working.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to change an m flag in a Translation Memory using the EQFCMM command.

Example

```
eqfcmm /mem=biotext /clear=yes /type=noconf
```

In this example, the m-flags that preceded the segments in the Translation Memory biotext are removed. You are not prompted with a confirmation message.

Specifying the quality of m-flagged segments

Purpose

Initial Translation Memory (ITM) specifying the quality of m-flagged segments (from command area) segments specifying the quality of m-flagged segments (from command area) Segments that were translated by machine are prefixed with an [m]. OpenTM2 provides a command to specify the quality of those translations.

You can specify a number between 0 and 101 for the quality. 101 stands for a perfect translation. The following translations, that is, the following aligned segments, get an m flag:

- All aligned segments if you specify 101
- The segments not reaching the specified quality level if you specify a number from 1 through 100
- No segments if you specify 0

A quality of 100 is decreased by certain findings:

- It is decreased by 1 if:
 - The source sentence contains less than 10 words and the target sentence more than 19
 - The source sentence contains more than 10 words but the target sentence is not even half as long as the source sentence
 - The source sentence contains more than 10 words but the target sentence is more than 1.5 times longer than the source sentence.
- It is decreased by the number of alignments that do not match 1:1 in the current alignment nor the five alignments before and after the current alignment.
- It is decreased by the number of differences between the source and target segment with respect to:
 - Numbers
 - Abbreviations
 - Words in uppercase characters only
 - The terms "F1", "F2", and so on
 - Formatting tags within a segment

Format

EQFITM specifying the quality of m-flagged segments commands EQFITM specifying the quality of m-flagged segments The following syntax diagram describes how to specify the quality of m-flagged segments with the EQFITM command. Note that lowercase letters in a parameter name are optional and need not be typed.

**EQFITM /MEm=memdb /FFiles= document_name (,document_name)
@list_file_name /MArket=markup
/SGmlmem=mem_file**

/SRcIng=source_language
/TGtIng=target_language
/SrcStartpath=source_startpath /TgtStartpath|/TS=target_startpath /TYpe= NOANA NOCONF
NOTM VISUAL PREPARE /QuaL=quality_level

Parameters

/MEm=memdb

memdb specifies the name of the **Initial Translation Memory** you want to work with.

/FIlEs=document_name

document_name specifies the name of a document you want to work with. The *document_name* specification can include the drive and directory where the file to work with is located.

If you specify several documents, enclose the document names in brackets and separate them by commas.

If you want to specify a list of document names in a list file, refer to this list file by preceding the list file name with the @ symbol (@ *list_file_name*). In the list file enclose the document names in brackets and separate them by commas.

If you do not specify the **FIlEs** option, all documents in the folder are selected.

/MArKup=markup

markup is the name of the markup table that the selected texts use (mandatory except for /TYPE=NOANA). See [Working with markup tables](#) for the names of the markup tables.

/SGmlMem=mem_file

mem_file is the name and path name for the external ITM. The ITM is in SGML format and can subsequently be imported into **OpenTM2** after you have checked it.

/SRcIng=source_language

source_language is the language of the original documents.

/TGtIng=target_language

target_language is the language of the translated documents.

/SrcStartpath=source_startpath

source_startpath is the path name of the source documents.

/TgtStartpath=target_startpath

target_startpath is the path name of the translated documents.

/TYpe=

One or more of the following:

- **NOANA**
Do not analyze the selected files because they have already been analyzed by **OpenTM2**.
- **NOCONF**
No confirmation message is displayed. This is useful if you do not want to be interrupted by progress messages while you are working.
- **NOTM**
Do not fill the internal **Translation Memory** (*memdb*) but the external **Translation Memory**. It is in SGML format and can be checked.
- **VISUAL** Creates the ITM and presents its contents on the screen for you to revise.
- **PREPARE** The source documents are related to their corresponding translations. The file pairs are prefixed with p.

/QuaL=quality_level

A number between 0 and 101 specifying the quality level of an alignment. If you omit this parameter, m-flags are added to all alignments unless you specify /TYPE=VISUAL.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example shows how to specify the quality of m-flagged segments using the EQFITM command.

Example

```
eqftm /fi=(us\al.txt, ger\alde.txt) /me=a1 /ma=EQFBOOK /SR=English (U.S.)  
/TG=German(national) /SS=D:\kbt\itm /TS=D:\kbt\itm /TY=(VISUAL) /QL=99
```

In this example, you request that all alignments with a quality level of less than 99 are m-flagged.

Changing the markup, target language and date of segments

Purpose

Translation Memory changing markup, target language and date of segments segments changing markup, target language and date markup language changing OpenTM2 enables you to change the markup language, the target language or the date of segments stored in a Translation Memory.

Format

EQFTMT commands EQFTMT The following syntax diagram describes how you change the markup, target language or date of segments with the EQFTMT command. Note that lowercase letters in a parameter name are optional and need not be typed.

```
EQFTMT  
/MEm=memdb  
/FromMarkup=frommu  
/ToMarkup=tomu  
/FromLang=fromlang /ToLang=tolang /DAte=newdate /DOC=document /TYpe=NOCONF
```

Parameters

/MEm=memdb

memdb specifies the name of the Translation Memory that you want to work with.

/FromMarkup=

frommu is the name of the markup language to be changed. The name can contain wildcard characters. For example, / FROMMARKUP=IBM* would change all segments with a markup starting with . If this parameter is omitted, the markup language of all segments is changed. This parameter can also be used to restrict the number of segments being changed when changing languages or segment dates. If specified without the /ToMarkup parameter only segments from the specified markup will be changed.

/ToMarkup=

tomu is the new markup language to be used for either the segments with markup language *frommu* or all segments if the /FROMMARKUP parameter is omitted.

Note: When the markup language of segments has been changed the tool performs an organize of the memory at the end of the processing. This ensures that the memory lookup works correctly after a change of the markup language.

/FromLang=

fromlang is the name of the target language to be changed. The name can contain wildcard characters. For example, / FROMLANG=Eng* would change all segments with a target language starting with Eng. If this parameter is omitted, the target language of all segments is changed.

/ToLang=

tolang is the new target language to be used for either the segments with target language *fromlang* or all segments if the /FROMLANG parameter is omitted.

/DAte=

newdate is the new date to be set for the segments. The format of newdate is YYYY-MM-DD HH:MM:SS. The time part (HH:MM:SS) is optional. You can also specify an asterisk ("*") as newdate, which instructs EQFTMT to use the current system date as segment date. For example, / DATE=2001-01-01 would change the segment date of all changed segments to the 1st January 2001.

/DOC=

using this parameter the changed segments can be restricted to segments from document document. The name can contain wildcard characters. For example, / DOC=A* would change all segments from documents starting with the letter A.

/TYpe=NOCONF

If you specify this parameter, no confirmation message is displayed. This is useful if you do not

want to be interrupted by progress messages while you are working.

If you do not specify this parameter, the "Change Markup and Target Language" window is displayed.

This window contains the following information:

- The name of the Translation Memory
- The number of the current segment
- The number of segments whose markup was changed
- The number of segments whose target language was changed
- The number of segments whose date was changed

At the end of the process, you get a completion message showing the number of segments processed and changed.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following examples show how to change the markup or language of segments using the EQFTMT command.

Example 1

```
eqftmt /mem=mymem /frommarkup=eqfami /tomarkup=eqfhtml4
```

In this example, the markup table of the **Translation Memory** mymem is changed from Ami Pro (eqfami) to HTML (eqfhtml4).

Example 2

```
eqftmt /mem=mymem /fromlang=English(U.S.) /tolang=German(national)
```

In this example, the target language is changed from English to German.

Example 3:

```
eqftmt /mem=mymem /doc=A* /date=*
```

In this example, the date of all segments from documents starting with the letter A in the Translation Memory mymem is changed to the current system date.

Removing segments with identical source and target strings

Purpose

OpenTM2 enables you to remove segments which have the same source and target string from a Translation Memory. Whitespace characters such as linefeed, carriage return and space are ignored when checking the source and target string. For security reasons the segments are not removed from the input Translation Memory but a new Translation Memory is created which contains all segments from the input Translation Memory except for the segments having the same source and target string.

Format

EQFTMCL commands EQFTMCL /MEm=*inmemdb* /OOut=*outmemdb*

Parameters

/MEm =*inmemdb*

/MEm *inmemdb* specifies the name of the Translation Memory that you want to use as input Translation Memory.

/OOut =*outmemdb*

outmemdb is the name of a new Translation Memory which is to receive all segments from the input Translation Memory except for the segments having the same source and target string. The Translation Memory *outmemdb* is created by EQFTMCL.

If you do not specify this parameter, the "Change Markup and Target Language" window is displayed. This window contains the following information:

- The name of the Translation Memory
- The number of the current segment
- The number of segments whose markup was changed
- The number of segments whose target language was changed

At the end of the process, you get a completion message showing the number of segments processed, the number of skipped segments, and the number of segments written to the output Translation Memory.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example show how to use the EQFTMCL command.

Enter the EQFTMCL command as follows:

Example

```
eqftmcl /mem=mymem /out=mymem-cleaned
```

In this example all segments contained in Translation Memory mymem are written to the new output Translation Memory mymem-cleaned except for segments having the same source and target string.

Removing inline tagging from an external memory

Purpose

OpenTM2 enables you to remove inline tagging from the segments in an external (= exported memory). For security reasons the segments are not updated inplace but a new memory is created containing the processed segments.

Format

```
EQFREMOVETAGS commandsEQFREMOVETAGS  
EQFREMOVETAGS  
/InMem=inmemdb /INTYPE=intype /OUTMEM=outmemdb /OUTTYPE=outtype  
– /MARKUP=markup
```

Parameters

/MEM =inmemdb

/MEM inmemdb specifies the fully qualified name of the external Translation Memory that you want to use as input Translation Memory.

/INTYPE=intype

intype specifies the format of the input memory and can be ASCII, ANSI, UTF16 or TMX

/OUTMEM =outmemdb

outmemdb is the fully qualified name of a new external Translation Memory which is to receive all segments from the input Translation Memory with the inline tagging removed.

/OUTTYPE=outtype

outtype specifies the format of the output memory and can be ASCII, ANSI, UTF16 or TMX or INTERNAL (to create an internal memory instead of an external)

/MARKUP=markup

markup specifies the markup to use for the recognition of inline tagging. The special value **RTF** can be used to remove pure RTF tagging from the segments, using the standard markup table EQFRFTF will also remove HTML like tags beside the pure RTF tags. If not specified the markup information of the segment itself is used for the inline tag recognition.

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example show how to use the EQFREMOVETAGS command.

Enter the EQFREMOVETAGS command as follows:

Example

```
eqfremovetags /inmem=c:\mymem.tmx /intype=TMX /outmem=c:\mymem-  
cleaned.exp
```

/outtype=UTF16 /markup=EQFRTF

In this example all segments contained in Translation Memory mymem are written to the new output Translation Memory mymem-cleaned except for segments having the same source and target string.

Showing the contents of exported folders

Purpose

OpenTM2 enables you to remove inline tagging from the segments in an external (= exported memory). For security reasons the segments are not updated inplace but a new memory is created containing the processed segments.

Format

SHOWFXP commandsSHOWFXP

SHOWFXP Folderfile /DETAILS

Parameters

folderfile

folderfile specifies the fully qualified file name of an exported folder (FXP)

/DETAILS

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*)

Examples

The following example show how to use the SHOWFXP command.

Enter the SHOWFXP command as follows:

Example

showfxp c:\folders\showme.fxp

In this example all segments contained in Translation Memory mymem are written to the new output Translation Memory mymem-cleaned except for segments having the same source and target string.

Restoring vital Translation Manager property files

Purpose

This tool may be used to restore vital OpenTM2 property files or to correct corrupted property files. Please contact TM-SUPPORT to get more information how to correct corrupted property files.

Format

EQFINST commandsEQFINST

EQFINST

Checking the folder history data

Purpose

OpenTM2 collects all processing steps and count information for a folder in a history log file. This history log file is the base for the creation of calculation reports. The tool CHKCALC can be used to verify that the folder history log file is not corrupted.

Format

CHKCALC commandsCHKCALC

CHKCALC /FLD=folder /HLOG=histlog /FXP=fxpfile /ALL

Parameters

/FLD =*folder*

folder specifies the name of a Translation Manager folder which is to be checked

/HLOG=*histlog*

histlog specified the fully qualified file name of a history log file to be checked

/FXP =*fxpfile*

fxpfile specified the fully qualified file name of an exported Translation Manager folder

/ALL

The /ALL switch tells the tool to check the history log information of all Translation Manager folders

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example show how to use the CHKCALC command.

Enter the CHKCALC command as follows:

Example

```
chkcalc /fld=showme  
In this example the history log information of folder showme is checked.
```

```
chkcalc /all
```

In this example the history log information of all folders is checked.

Correcting the drive letter information of Translation Manager files

Purpose

Many OpenTM2 files contain drive letter information. When these files are copied from one machine to another or to another drive the files can not be used anymore and are shown as marked as defect in the OpenTM2 list windows. The tool EQFADL corrects the drive letter settings of the specified file so that the files can be processed as usual.

Format

EQFADL commands EQFADL

EQFADL /FOL=*folder* /MEM=*mem* /DIC=*dictionary*

Parameters

• **/FOL =*folder***

folder specifies the name of a Translation Manager folder whose drive letter settings will be corrected

• **/MEM =*mem***

mem specifies the name of a Translation Memory whose drive letter settings will be corrected

• **/DIC =*dictionary***

dictionary specified the name of a dictioanry whose drive letter settings will be corrected

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example show how to use the EQFADL command.

Enter the EQFADL command as follows:

...

Converting TM SGML memory databases into TMX format

Purpose

OpenTM2 enables you to remove inline tagging from the segments in an external (= exported memory). For security reasons the segments are not updated inplace but a new memory is created containing the processed segments.

Format

EXP2TMX commands**EXP2TMX**

EXP2TMX InMEM -- OUTMEM -- /INMODE=im -- /OUTMODE=om

Parameters

- **inmem**
is the fully qualified file name of the input memory in TM SGML format (This name can contain wildcard characters)
- **outmem**
is the fully qualified file name of the output memory in TMX format, when no output memory name is specified the input memory name is used as output memory name with the extension changed to TMX
- **/INMODE =im**
im specifies the encoding of the input memory and can be UTF16, ASCII, ANSI or the number of a codepage. The input mode has to be specified only when the memory is not encoded in UTF16 and does not contain the <codepage> tag in the header
- **/OUTMODE=om**
om specifies the encoding of the output memory and can be UTF8 or UTF16, When not specified UTF8 is used as default

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example show how to use the EXP2TMX command.

Enter the EXP2TMX command as follows:

...

In this example all EXP memory databases located in folder c:\memory are converted into the TMX format, the output memory name will be the input memory name with an extension of TMX and the encoding of the output memory will be UTF8.

Converting external Memory databases from the TMX format into the EXP format

Purpose

For the memory interchange with other tools the memory interchange format TMX is widely used. The TMX2EXP tool can be used to convert TMX Translation Memory databases into the OpenTM2 specific EXP format.

Format

TMX2EXP commands**TMX2EXP**

TMX2EXP InMEM -- OUTMEM -- /OUTMODE=om -- /CLEANRTF -- /SOURCESOURCE -- /MARKUP=mu

Parameters

- **inmem**
inmem specifies the name of the external Translation Memory in EXP format that you want to convert, the name may contain wildcards (e.g. C:\MEMS*.EXP) to convert a group of files in a single run
- **outmem**
outmem is the name of the converted Translation Memory in EXP format. If no name is specified the name of the input memory is used with the file extension changed to .EXP.
- **/OUTMODE=om or /OM=om**
om specifies the encoding of the output memory and can be UTF16, ASCII, or ANSI.

- **/CLEANRTF**
if specified any RTF inline tags are removed from the segments of the memory
- **/SOURCESOURCE or /SS**
if specified the source of the memory matches is also used as the target text thus creating a source-source memory
- **/MARKUP=*mu* or /MA=*mu***
if specified the source of the memory matches is also used as the target text thus creating a source-source memory

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example show how to use the TMX2EXP command.

Enter the TMX2EXP command as follows:

...

In this example all Translation Memory databases with an extension of .TMX will be converted into the EXP format and the encoding of the converted memories will be UTF-16, all RTF tagging contained in the segments will be removed.

Changing the type of an exported folder

Purpose

Exported OpenTM2 folder can be master folders, child folders or standard folders. When exporting a master folder it is automatically exported as a child folder when not specified otherwise. The tool CHANGEFXP can be used to change the type of the exported folder without requiring a new export of the folder.

Format

ChTypeExportFolder commandsChTypeExportFolder

CHANGEFXP *folderfile* - /TYPE= **MASTER CHILD STANDARD** - /PW=*password*

Parameters

- **folderfile**
folderfile is the fully qualified name of the exported folder, if no file extension is specified ".FXP" is used as default
- **/TYPE=*type***
type is the new type of the folder and can be MASTER, CHILD, or STANDARD
- **/PW =*password***
password specifies the folder password for MASTER or CHILD folders

Note:

You can type the commands in uppercase, lowercase, or in mixed-case letters. You may not use any wildcard characters, such as an asterisk (*).

Examples

The following example show how to use the CHANGEFXP command

Enter the CHANGEFXP command as follows:

...

Programming interfaces

Application programming interface for adding editors

OpenTM2 provides an application programming interface (API) that lets you use various editors as translation editors. Using this API the editor can access all functions required for a translation, namely the **Translation Memory**, the automatic dictionary lookup, and the dictionary lookup dialog. OpenTM2 prepares the "Dictionary" and "Translation Memory" windows, establishes the communication links, handles all error conditions, and prepares and accesses the dictionaries and

Translation Memory databases. The editor must provide the end-user interface to access the provided services and handle the retrieved data.

All API functions are provided as a dynamic-link library (DLL).

An editor that can be used as a translation editor must meet the following requirements:

- Run as a Presentation Manager^(R) application. A VIOwindowed application is not sufficient.
- Be programmable.
- Be able to access programs and DLLs written in C for multithread environments.
- Be able to recognize specific tags and extract and decompose text according to this information.

The following sections describe the data types used by the API interface, possible error conditions, and the individual API calls for the interface provided by **OpenTM2**.

Data types

The editor must use the following structure to communicate with **OpenTM2**. The C interface binding is available in the file EQFTWBS.H.

```
typedef struct _STEQFSTRUCT
{
    HWND hwndEdit ; /* handle of editor window */
    CHAR szSemaphore [EQF_NAME]; /* space for the semaphore name */
    HWND hwndEQFPropWnd; /* handle of proposal window */
    HWND hwndEQFDictWnd; /* handle of dictionary window */
    USHORT usIndustryCode; /* industry code */
    CHAR szProjPath [EQF_NAME]; /* path of file to be translated*/
    CHAR szFileName [EQF_NAME]; /* currently transl. file */
    RECTL rectIEQFPropWnd; /* coordinates of proposal wnd */
    RECTL rectIEQFDictWnd; /* coordinates of dictionary wnd*/
    SHORT sOS2; /* Error code */
} STEQFSTRUCT;
```

Return codes

The following list contains all return codes provided by **OpenTM2**. If an operating-system error is found, the EQFERR_SYSTEM is set and the extended return code is updated in the line stEQFStruct.sOS2.

- **EQFERR_TM_ACCESS**
The **Translation Memory** could not be accessed.
- **EQFERR_DICT_ACCESS**
The dictionary or the dictionary lookup program could not be accessed.
- **EQF_OKAY**
The request completed successfully.
- **EQFERR_INIT**
The system must first be initialized.
- **EQFERR_CLOSE_DICT**
An error occurred during the closing of the dictionary.
- **EQFERR_CLOSE_TM**
An error occurred during the closing of the **Translation Memory**.
- **EQFERR_ENTRY_NOT_AVAIL**
The selected proposal is not available.
- **EQFERR_DISK_FULL**
OpenTM2 detected that the disk is full.
- **EQFERR_TM_NOT_ACTIVE**
The **Translation Memory** is not active.
- **EQFERR_SEG_EMPTY**
The passed segment was empty and therefore was not stored in the **Translation Memory**.
- **EQFERR_TM_CORRUPTED**
The **Translation Memory** is corrupted.

- **EQFERR_SEG_NOT_FOUND**
The specified segment was not found.
- **EQFERR_DICTLOOK_NOT_FOUND**
The dictionary lookup dialog could not be loaded.
- **EQFERR_DICT_LOOKUP_PENDING**
The dictionary lookup request is pending.
- **EQFERR_NO_ENTRY_AVAIL**
The dictionary entry is not available.
- **EQFERR_SYSTEM**
A system error occurred.

API calls

The following sections describe the individual API calls for the interface provided by **OpenTM2**.
The following calls are available:

Call...	described on page...
EQFCLEAR	...
EQFCLOSE	...
EQFCONVERTFILENAMES	...
EQFDELSEG	...
EQFDICTLOOK	...
EQFFILECONVERSIONEX	...
EQFGETDICT	...
EQFGETDOCFORMAT	...
EQFGETPROP	...
EQFGETSENUM	...
EQFGETSOURCELANG	...
EQFGETTARGETLANG	...
EQFINIT	...
EQFQUERYEXITINFO	...
EQFSAVESEG	...
EQFSEGFILECONVERTASCII2UNICODE	...
EQFSEGFILECONVERTUNICODE2ASCII	...
EQFTRANSSEG	...
EQFWORDCNTPERSEG	...
EQFWRITEHISTLOG	...

EQFCLEAR

Purpose

EQFCLEAR programming interface calls **EQFCLEAR** *EQFCLEAR* resets or clears the information stored.

Format

EQFCLEAR(usFlag)

Parameters

- *usFlag(USHORT)*
Can be either of the following:
 - **EQFF_NODICTWND**
The "Dictionary" window is hidden.
 - **EQFF_NOPRDPWND**
The "Proposals" window is hidden.

Return codes

- **EQF_OKAY**
The request completed successfully.
- **EQFERR_INIT**
The system must first be initialized.

Remarks

This call is used to initialize the buffers and clear the "Dictionary" and "Translation Memory" windows after a new document is loaded.

EQFCLOSE

Purpose

EQFCLOSE programming interface calls EQFCLOSE. EQFCLOSE closes the session with OpenTM2.

Format

EQFCLOSE(fShutdown)

Parameters

- *fShutdown*

Can be either of the following:

- **EQF_CLOSE_STANDBY**

The services session is closed, the services remain active.

- **EQF_CLOSE_EXIT**

The services are closed and destroyed.

Return codes

- **EQF_OKAY**

The request completed successfully.

- **EQFERR_INIT**

The system must first be initialized.

- **EQFERR_CLOSE_DICT**

An error occurred during the closing of the dictionary.

- **EQFERR_CLOSE_TM**

An error occurred during the closing of the Translation Memory.

- **EQFERR_SYSTEM**

A system error occurred.

Remarks

This call must be the last OpenTM2 call, implicitly issued by OpenTM2.

EQFCONVERTFILENAMES

Purpose

EQFCONVERTFILENAMES programming interface calls EQFCONVERTFILENAMES. EQFCONVERTFILENAMES converts long file names into short file names, and vice versa.

If the long file name is an empty string, the long file name is created from the short file name, and vice versa. If the short file name meets the 8.3 DOS naming conventions, the long file name is returned as a null pointer.

Format

EQFCONVERTFILENAMES(pszFolder,pszLongFileName,pszShortFileName)

Parameters

- *pszFolder(PSTRING) — input*

The name of the folder with path information, for example

<folder_drive>:\eqf\<folder_name>.f00. <folder_name> can be extracted from pSegTarget or pSegSource as defined in eqf_xstart.

- *pszLongFileName(PSTRING) — input or output*

The long file name without path information. It is used to get the short file name. If *pszLongFileName==NULL*, *pszLongFileName* is output.

- *pszShortFileName(PSTRING) — input or output*

The short file name (8.3 DOS naming convention) without path information. It is used to get the long file name. If *pszShortFileName==NULL*, *pszShortFileName* is output.

Return codes

A OpenTM2 return code as defined in the file OS2TOWIN.H. A return code of null indicates

successful processing.

Remarks

If a long file name is to be created from a short file name and the result is an empty string for *pszLongFileName*, the short file name applies to the 8.3 naming conventions.

Notes

Either *pszLongFileName* or *pszShortFileName* must be an empty string. The non-empty string must be a valid file name, otherwise an error is recorded.

EQFDELSEG

Purpose

EQFDELSEG programming interface calls EQFDELSEG. EQFDELSEG deletes the specified segment from the **Translation Memory**, together with its information.

Format

EQFDELSEG(pszBuffer1,pszBuffer2,usSegNum)

Parameters

- **pszBuffer1(PSTRING) — input**

The buffer for the source segment to be deleted. It must have a length of `EQF_BUFFERLEN`. `EQF_BUFFERLEN` is defined in the file `EQFTWBS.H`.

- **pszBuffer2(PSTRING) — input**

The buffer for the corresponding translation to be deleted. It must have a length of `EQF_BUFFERLEN`. `EQF_BUFFERLEN` is defined in the file `EQFTWBS.H`.

- **usSegNum(USHORT) — input**

The segment number.

Return codes

- **EQFERR_SEG_NOT_FOUND**

The specified segment was not found.

- **EQFERR_TM_ACCESS**

The **Translation Memory** could not be accessed.

- **EQFERR_TM_CORRUPTED**

The **Translation Memory** is corrupted.

- **EQF_OKAY**

The request completed successfully.

- **EQFERR_INIT**

The system must first be initialized.

Remarks

This call is useful if parts of combined segments are already translated. These parts are now meaningless and can therefore be deleted from the **Translation Memory**.

EQFDICTLOOK

Purpose

EQFDICTLOOK programming interface calls EQFDICTLOOK. EQFDICTLOOK invokes the dictionary lookup dialog.

Format

EQFDICTLOOK(pszBuffer1,pszBuffer2,usCursorPos, fSource)

Parameters

- **pszBuffer1(PSTRING) — input**

The buffer for the active segment. It must have a length of `EQF_BUFFERLEN`. `EQF_BUFFERLEN` is defined in the file `EQFTWBS.H`.

- **pszBuffer2(PSTRING) — input**

The buffer for the marked area. It must have a length of `EQF_BUFFERLEN`. `EQF_BUFFERLEN` is defined in the file `EQFTWBS.H`.

- **usCursorPos(USHORT) — output**

The position of the input cursor.

- ***fSource — output***
Determines whether the term looked up is in the source or target language (not used in the current **OpenTM2** version).

Return codes

- **EQFERR_DICTLOOK_NOT_FOUND**
The dictionary lookup dialog could not be loaded.
- **EQF_OKAY**
The dictionary term is selected and copied into the provided buffer.
- **EQFERR_INIT**
The system must first be initialized.
- **EQFERR_DICT_LOOKUP_PENDING**
The dictionary lookup request is pending.
- **EQFERR_NO_ENTRY_AVAIL**
The dictionary entry is not available.

Remarks

EQFERR_DICT_LOOKUP_PENDING indicates that a dictionary lookup is active. After selecting an entry or leaving the dictionary lookup dialog, the return code is reset to either EQF_OKAY or EQF_NO_ENTRY_AVAIL. From an editor's point of view, this call is handled in the same way as EQFGETDICT (see page [EQFGETDICT](#) [EQFGETDICT](#)).

EQFILECONVERSIONEX

Purpose

EQFILECONVERSIONEX programming interface calls **EQFILECONVERSIONEX**

EQFILECONVERSIONEX is a helper function for user exits which require the files to be converted.

The new API function gives the possibility

- to convert an ASCII file into ANSI (EQF_ASCII2ANSI)
- to convert an ANSI file into ASCII (EQF_ANSI2ASCII)
- to convert an ASCII file into UTF8 (EQF_ASCII2UTF8)
- to convert an UTF8 file into ASCII (EQF_UTF82ASCII)
- to convert an ASCII file into UTF16 (EQF_ASCII2UTF16)
- to convert an UTF16 file into ASCII (EQF_UTF162ASCII)
- to convert an ANSI file into UTF8 (EQF_ANSI2UTF8)
- to convert an UTF8 file into ANSI (EQF_UTF82ANSI)
- to convert an ANSI file into UTF16 (EQF_ANSI2UTF16)
- to convert an UTF16 file into ANSI (EQF_UTF162ANSI)
- to convert an UTF8 file into UTF16 (EQF_UTF82UTF16)
- to convert an UTF16 file into UTF8 (EQF_UTF162UTF8)

Format

EQFILECONVERSIONEX(pszInFile,pszOutFile,pszLanguage,usConversionType)

Parameters

- ***pszInFile(PSZ) — input***
the fully qualified filename of the input file. or as defined in .
- ***pszOutFile(PSZ) — input***
the fully qualified filename of the output file.
- ***pszLanguage(PSZ) — input***
the language of the file (e.g. it can be retrieved with EQFGETSOURCELANG/EQFGETTARGETLANG).
- ***usConversionType(USHORT) — input***
identifier of type of conversion: ASCII2ANSI, ANSI2ASCII
 - EQF_ASCII2ANSI
 - EQF_ANSI2ASCII
 - EQF_ASCII2UTF8
 - EQF_UTF82ASCII
 - EQF_ASCII2UTF16
 - EQF_UTF162ASCII
 - EQF_ANSI2UTF8
 - EQF_UTF82ANSI
 - EQF_ANSI2UTF16

- EQF_UTF162ANSI
- EQF_UTF82UTF16
- EQF_UTF162UTF8
- usReturn(USHORT) — **output**
 - EQFRC_OK successfully completed
 - EQFS_FILE_OPEN_FAILED file cannot be opened
 - ERROR_STORAGE allocation of memory failed
 - ERROR_FILE_INVALID_DATA file contains data that cannot be converted
 - EQFRS_INVALID_PARM in all other cases of error

Return codes

- EQFRC_OK successfully completed
- EQFS_FILE_OPEN_FAILED file cannot be opened
- ERROR_STORAGE allocation of memory failed
- ERROR_FILE_INVALID_DATA file contains data that cannot be converted
- EQFRS_INVALID_PARM in all other cases of error

Remarks

If the file pszOutFile exists already, it is overwritten.

The API EQFFILECONVERSION is not available any more in TM6.0.2. It has been replaced by the new API EQFFILECONVERSIONEX.

The pszInFile is converted according to the conversion type and written as the file pszOutFile. Output file and input file should be different files.

The input language is used to determine the ASCII and ANSI codepage for the conversion. Inside TM, exactly one ASCII /one ANSI codepage is attached to each possible language. The input language must be a valid TM source or target language.

If the language is NULL, the default target language of the system preferences is used for conversion.

If EQF_ASCII2ANSI is specified, it is assumed that the input file is in ASCII. If EQF_ANSI2ASCII is specified, it is assumed that the input file is in ANSI. If EQF_UTF162ANSI or EQF_UTF162ASCII or EQF_UTF162UTF8 is specified, the input file is checked for the byte order mark. For UTF16 files, a byte-order-mark is required. If the input file does not contain such a mark, ERROR_FILE_INVALID_DATA is returned.

For UTF8 input files, a byte-order-mark is accepted, however it is not required. UTF8 output files are written without a byte-order-mark.

If the input file contains characters which are not valid in the codepage of the input language, the API EQFFILECONVERSIONEX may fail with the error return ERROR_FILE_INVALID_DATA. EQFRS_INVALID_PARM is returned as error code if usConversionType is invalid.

Examples

Example

```
CHAR szInFile[145];
CHAR szOutFile[145];
CHAR szLanguage[20];
USHORT usRC = 0;
strcpy(szOutFile, "d:\temp\b.tst");
strcpy(szInFile, "d:\input\b.tst");
strcpy(szLanguage, "English(U.S)");

usRC = EQFILECONVERSIONEX( szInFile, szOutFile, szLanguage, EQF_ASCII2ANSI );
```

EQFGETDICT

Purpose

EQFGETDICT programming interface calls **EQFGETDICT**. **EQFGETDICT** retrieves the selected dictionary word and copies it into the provided buffer.

EQF_UP or **EQF_DOWN** scrolls the contents of the "Dictionary" window in the selected direction, if possible. **EQF_LOOKUP** can be used to retrieve the selected dictionary lookup term. The appropriate return code is set if the dictionary lookup is pending or no term is selected. **EQF_UP**, **EQF_DOWN**, and **EQF_LOOKUP** are defined in the file **EQFTWBS.H**.

Format

EQFGETDICT(*usNum,pszBuffer*)

Parameters

- ***usNum(USHORT)* — input**
The number of the selected dictionary word (0...9, EQF_UP, EQF_DOWN, EQF_LOOKUP).
- ***pszBuffer(PSTRING)* — output**
The buffer for the dictionary word. It must have a length of EQF_BUFFERLEN. EQF_BUFFERLEN is defined in the file EQFTWBS.H.

Return codes

- **EQFERR_ENTRY_NOT_AVAIL**
The selected dictionary entry is not available.
- **EQF_OKAY**
The selected dictionary term is available and copied into the provided buffer.
- **EQFERR_INIT**
The system must first be initialized.

Remarks

If the selected dictionary word is not available, a warning message is issued.

EQFGETDOCFORMAT

Purpose

EQFGETDOCFORMAT programming interface calls **EQFGETDOCFORMAT**.
EQFGETDOCFORMAT retrieves the format (markup language) of the specified document.

Format

EQFGETDOCFORMAT(*pszFolder,pszFileName,pszFormat*)

Parameters

- ***pszFolder(PSTRING)* — input**
The name of the folder with path information, for example
<folder_drive>:\eqf\<folder_name>.f00. *<folder_name>* can be extracted from pSegTarget or pSegSource as defined in eqf_xstart.
- ***pszFileName(PSTRING)* — input**
The short file name (8.3 DOS naming convention) without path information.
- ***pszFormat(PSTRING)* — output**
The format (markup language) of the specified document.

EQFGETPROP

Purpose

EQFGETPROP programming interface calls **EQFGETPROP**. **EQFGETPROP** retrieves the selected proposal and copies it to the provided buffer.

EQF_UP or **EQF_DOWN** scrolls the contents of the "Translation Memory" window in the selected direction, if possible. **EQF_UP** and **EQF_DOWN** are defined in the file EQFTWBS.H.

Format

EQFGETPROP(*usNum,pszBuffer,pusLevel*)

Parameters

- ***usNum(USHORT)* — input**
The number of the selected proposal or match (0...9, EQF_UP, EQF_DOWN).
- ***pszBuffer(PSTRING)* — output**
The buffer for the **Translation Memory** proposals. It must have a length of EQF_BUFFERLEN. EQF_BUFFERLEN is defined in the file EQFTWBS.H.
- ***pusLevel(PUSHORT)***
The pointer to the variable for the return match level.

Return codes

- **EQFERR_ENTRY_NOT_AVAIL**

The selected proposal is not available.

- **EQF_OKAY**
The selected proposal is available and copied into the provided buffer.
- **EQFERR_INIT**
The system must first be initialized.

Remarks

If the selected proposal is not available, a warning message is issued and the appropriate return code is set.

EQFGETSEGNUM

Purpose

EQFGETSEGNUM programming interface calls EQFGETSEGNUM. EQFGETSEGNUM retrieves the segment number of the currently selected proposal (the segment that was used before by the EQFGETPROP call).

Format

EQFGETSEGNUM(pulSegNum)

Parameters

- **pulSegNum(PULONG) — output**
The pointer to the ULONG variable receiving the segment number.

Return codes

One of the values listed in [Return codes](#).

Remarks

You can use the retrieved segment number, for example, as input parameter with the EQFDELSEG call.

EQFGETSOURCELANG

Purpose

EQFGETSOURCELANG programming interface calls EQFGETSOURCELANG. EQFGETSOURCELANG retrieves the source language of the specified document.

Format

EQFGETSOURCELANG(pszFolder,pszFileName,pszSrcLang)

Parameters

- **pszFolder(PSTRING) — input**
The name of the folder with path information, for example
`<folder_drive>:\eqf\<folder_name>.f00`. `<folder_name>` can be extracted from pSegTarget or pSegSource as defined in eqf_xstart.
- **pszFileName(PSTRING) — input**
The short file name (8.3 DOS naming convention) without path information.
- **pszSrcLang(PSTRING) — output**
The source language.

EQFGETTARGETLANG

Purpose

EQFGETTARGETLANG programming interface calls EQFGETTARGETLANG. EQFGETTARGETLANG retrieves the target language of the specified document.

Format

EQFGETTARGETLANG(pszFolder,pszFileName,pszTrgLang)

Parameters

- **pszFolder(PSTRING) — input**
The name of the folder with path information, for example
`<folder_drive>:\eqf\<folder_name>.f00`. `<folder_name>` can be extracted from pSegTarget or pSegSource as defined in eqf_xstart.
- **pszFileName(PSTRING) — input or output**
The short file name (8.3 DOS naming convention) without path information.

- *pszSrcLang(PSTRING) — output*
The target language.

EQFINIT

Purpose

EQFINIT programming interface callsEQFINIT. EQFINIT initializes **OpenTM2** for use by an editor. This means, it creates the "Dictionary" and "Translation Memory" windows, establishes the communication links, attaches the **Translation Memory** and dictionaries, and allocates the internal structures required by **OpenTM2**.

Format

EQFINIT(pstEQFStruct,pszTranslationMemoryDatabases,pszUserDictionaries)

Parameters

- *pstEQFStruct(PSTEQFSTRUCT) — input*
The number of sentences (0...9).
- *pszTranslationMemoryDatabases*
The file name of the Translation Memory databases.
- *pszUserDictionaries*
The name of the user-supplied dictionaries.

Return codes

- **EQFERR_TM_ACCESS**
The **Translation Memory** could not be accessed.
- **EQFERR_TM_CORRUPTED**
The **Translation Memory** is corrupted.
- **EQFERR_DICT_ACCESS**
The dictionary or the dictionary lookup program could not be accessed.
- **EQF_OKAY**
The request completed successfully.
- **EQFERR_SYSTEM**
A system error occurred.

Remarks

The application must set the initial values for the position and size of the "Dictionary" and "Translation Memory" windows. If nothing is specified, the default values are used. If a problem occurs, a warning message is issued and the appropriate return code is set.

Notes

This call is implicitly issued by **OpenTM2** and only listed for completeness reasons.

EQFQUERYEXITINFO

Purpose

EQFQUERYEXITINFO programming interface callsEQFQUERYEXITINFO. The entry point **EQFQUERYEXITINFO** in **QUERYEXIT_ADDFILES** mode is called by **OpenTM2** during folder export when a markup table having a user exit is added to the exported folder.

If the user exit requires other files beside the markup table (.TBL) file, the user exit DLL and the .markup table control file (.CHR) to be exported and imported with the folder it should place a list of these files in the supplied buffer area.

The list of files is a comma separated list of file names terminated by a null character (C string syntax).

The file names may not contain wildcard characters.

All files are specified with their relative path in the \EQF directory.

Files not located in the \EQF directory cannot be exported and imported using folder import.

Example:

The file list "TABLE\ADDFILE.CHR,WIN\MYDLL.DLL,WIN\LOCALE\XYZ.CNV" will export the

files \EQF\TABLE\ADDFILE.CHR, \EQF\WIN\MYDLL.DLL and the file \EQFWIN\LOCALE\YXZ.CNV in the exported folder. OpenTM2 versions prior to TP603 will only import the files contained in the \EQF\TABLE directory, files in other directories will be ignored.

If the user exit places a list of additional files in the supplied buffer it should return a return code of zero all other values are assumed to be error codes.

In the future there will be other modes of the entry point EQFQUERYEXITINFO, so the requested mode should be checked by the user exit.

Format

EQFQUERYEXITINFO(*pszTagTable,usMode,pszBuffer,usBufLen*)

Parameters

- ***pszTagTable(PSZ) — input***
The name of the active tag table; e.g. "IBMHTM32"
- ***usMode(USHORT) — input***
Mode of the function, currently on "QUERYEXIT_ADDFILES" is being used
- ***pszBuffer(PSZ) — input***
Points to a buffer which will receive the list of additional markup table files
- ***usBufLen(USHORT)— input***
Length of the supplied buffer area in number of bytes

Return codes

- **USHORT (zero = function completed successfully)**

EXAMPLE:

```
USHORT APIENTRY16 EQFQUERYEXITINFO
(
    PSZ pszTagTable, // name of the markup table, e.g. "IBMHTM32"
    USHORT usMode, // type of information being queried
    PSZ pszBuffer, // buffer area receiving the information
    returned by the exit

    USHORT usBufLen // length of buffer area
)
{
    switch( usMode )
    {
        case QUERYEXIT_ADDFILES:
            strcpy( pszBuffer, "TABLE\MYINFO.CTL,WIN\MYDLL.DLL" );
            break;
        default:
            usRC = 1; // mode is not supported by user exit
    } /* endswitch */
} /* end of function EQFQUERYEXITINFO */
In this sample the files "\EQFTABLE\MYINFO.CTL" and "\EQFWIN\MYDLL.DLL"
are exported within the exported folder package.
```

EQFSAVESEG

Purpose

EQFSAVESEG programming interface calls **EQFSAVESEG**. EQFSAVESEG saves the passed segment information in the **Translation Memory**.

Format

EQFSAVESEG(*pszBuffer1,pszBuffer2,usSegNum*)

Parameters

- ***pszBuffer1(PSTRING) — input***
The buffer for the source segment. It must have a length of **EQF_BUFFERLEN**. **EQF_BUFFERLEN** is defined in the file **EQFTWBS.H**.
- ***pszBuffer2(PSTRING) — input***
The buffer for the translated segment. It must have a length of **EQF_BUFFERLEN**. **EQF_BUFFERLEN** is defined in the file **EQFTWBS.H**.

- ***usSegNum(USHORT) — input***
The segment number.

Return codes

- **EQFERR_DISK_FULL**
OpenTM2 detected that the disk is full.
- **EQFERR_TM_NOT_ACTIVE**
The Translation Memory is not active.
- **EQFERR_SEG_EMPTY**
The passed segment was empty and therefore was not stored in the Translation Memory.
- **EQF_OKAY**
The dictionary term is selected and copied into the provided buffer.
- **EQFERR_INIT**
The system must first be initialized.

Remarks

The editor must ensure that only correct data is saved in the Translation Memory. This means that the application must first check the spelling of the data.

EQFSEGFILECONVERTASCII2UNICODE

Purpose

EQFSEGFILECONVERTASCII2- UNICODE programming interface callsEQFSEGFILECONVERTASCII2- UNICODE EQFSEGFILECONVERTASCII2UNICODE gives the possibility to convert the segmented ASCII file to UTF16-Unicode (EQFSEGFILECONVERTASCII2UNICODE).

EQFSegFileConvertASCII2Unicode are helper functions for user exits which require the segmented files to be in ASCII whereas OpenTM2 expects the segmented files to be saved in Unicode.

The pszInFile is converted from ASCII to Unicode and written as the file pszOutFile. If the file pszOutFile already exists, it is overwritten. Only files which are correctly segmented, can be converted with this API.

Format

EQFSEGFILECONVERTASCII2UNICODE(pszInFile,pszOutFile,)

Parameters

- ***pszInFile(PSZ) — input***
The fully qualified filename of a segmented file in ASCII format which should be converted .
- ***pszOutFile(PSZ) — input***
the fully qualified filename of the file to which pszInFile should be converted.
- ***usReturn(USHORT) — output***

Return codes

- **EQFRC_OK**
successfully completed
- **ERROR_FILE_OPEN_FAILED**
file read error
- **ERROR_STORAGE**
allocation of memory failed.
- **ERROR_FILE_INVALID_DATA**
segmentation of file is erroneous
- **EQFRS_INVALID_PARM**
table cannot be accessed

Remarks

If the file pszOutFile exists already, it is overwritten.

EQFSEGFILECONVERTUNICODE2ASCII

Purpose

EQFSEGFILECONVERTUNICODE2ASCII programming interface callsEQFSEGFILECONVERTUNICODE2ASCII. EQFSEGFILECONVERTUNICODE2ASCII gives the possibility to convert the segmented UTF16 -Unicode file to ASCII (EQFSEGFILECONVERTUNICODE2ASCII). EQFSEGFILECONVERTUNICODE2ASCII are helper functions for user exits which require the segmented files to be in ASCII whereas OpenTM2 expects the segmented files to be saved in Unicode.

The pszInFile is converted from Unicode to ASCII and written as the file pszOutFile. If the file pszOutFile already exists, it is overwritten. Only files which are correctly segmented, can be converted with this API.

Format

EQFSEGFILECONVERUNICODE2ASCII(pszInFile,pszOutFile,)

Parameters

- **pszInFile(PSZ) — input**
The fully qualified filename of a segmented file in UTF16 Unicode format which should be converted .
- **pszOutFile(PSZ) — input**
the fully qualified filename of the file to which pszInFile should be converted.
- **usReturn(USHORT) — output**

Return codes

- **EQFRC_OK**
successfully completed
- **ERROR_FILE_OPEN_FAILED**
file read error
- **ERROR_STORAGE**
allocation of memory failed.
- **ERROR_FILE_INVALID_DATA**
segmentation of file is erraneous
- **EQFRS_INVALID_PARM**
table cannot be accessed

Remarks

If the file pszOutFile exists already, it is overwritten.

EQFTRANSSEG

Purpose

EQFTRANSSEG programming interface callsEQFTRANSSEG. EQFTRANSSEG retrieves the information available for the current segment and puts it into the internal waiting list.

OpenTM2 handles the layout and scrolling of the "Dictionary" and "Translation Memory" windows and the selection of entries.

Format

EQFTRANSSEG(pszBuffer,usSegNum,fShow,fFlags)

Parameters

- **pszBuffer(PSTRING) — input**
The buffer for the source segment. It must have a length of EQF_BUFFERLEN. EQF_BUFFERLEN is defined in the file EQFTWBS . H.
- **usSegNum(USHORT) — input**
The segment number.
- **fShow(BOOL) — input**
Determines whether the segment must immediately be displayed in the "Dictionary" or "Translation Memory" window:
 - **TRUE**

Put the segment into the "Dictionary" or "**Translation Memory**" window.

- **FALSE**

Use the segment information as sentence.

- ***fFlags(FLAG) — input***

Determines what is displayed:

- **EQF_NODICTWND**

No "Dictionary" window is displayed.

- **EQF_NOPROPWND**

No "**Translation Memory**" window is displayed.

- **EQF_NOAUTODICT**

The automatic dictionary lookup is disabled.

Return codes

- **EQFERR_DISK_FULL**

OpenTM2 detected that the disk is full.

- **EQFERR_TM_CORRUPTED**

The **Translation Memory** is corrupted.

- **EQFERR_TM_ACCESS**

The **Translation Memory** could not be accessed.

- **EQFERR_DICT_ACCESS**

The dictionary or the dictionary lookup program could not be accessed.

- **EQF_OKAY**

The request completed successfully.

- **EQFERR_INIT**

The system must first be initialized.

Remarks

If *fShow* is set to FALSE, the success indicator is immediately set to TRUE. In addition, the sentence is treated as a sentence and processed in the background. Any error information produced during background processing is stored and displayed when this segment is displayed.

If *fShow* is set to TRUE, this call first checks if the segment information is already prepared and can be immediately retrieved. If this is not the case, it is processed in the foreground.

The EQF_NOAUTODICT flag is used to determine if the dynamic dictionary lookup, which consumes a lot of performance, should be skipped.

EQWORDCNTPERSEG

Purpose

EQWORDCNTPERSEG programming interface calls **EQWORDCNTPERSEG**

EQWORDCNTPERSEG counts the number of words and markup tags in the specified segment using the specified language and markup. To count the number of words in a document, the words must be counted segment by segment.

Format

EQWORDCNTPERSEG(pszSeg,pszLang,pszFormat, pulResult,pulMarkup)

Parameters

- ***pszSeg(PSTRING) — input***

The segment of which the number of words and markup tags must be counted.

- ***pszLang(PSTRING) — input***

The source or target language as provided by **EQFGETSOURCELANG** (see page [EQFGETSOURCELANG](#)) or **EQFGETTARGETLANG** (see page [EQFGETTARGETLANG](#)).

- ***pszFormat(PSTRING) — input***

The format of the document as provided by **EQFGETDOCFORMAT** (see page [EQFGETDOCFORMAT](#)).

- ***pulResult(PULONG) — output***

The result of word counting.

- **pulMarkUp(PULONG) — output**
The result of markup-tag counting.

EQFWRITEHISTLOG

Purpose

EQFWRITEHISTLOG programming interface calls **EQFWRITEHISTLOG**. **EQFWRITEHISTLOG** writes the word-counting information to the history log file of the specified folder. The word-counting information for the entire document is needed.

Format

EQFWRITEHISTLOG(pszFoObjName,pszDocName,pszHistLogApi)

Parameters

- **pszFolder(PSTRING) — input**
The name of the folder with path information, for example
<folder_drive>:\eqf\<folder_name>.f00. <folder_name> can be extracted from pSegTarget or pSegSource as defined in eqf_xstart.
- **pszFileName(PSTRING) — input**
The short file name (8.3 DOS naming convention) without path information.
- **pszHistLogApi(PAPIDOCSAVEHIST) — input**
The structure of the history log file:

```
typedef struct _APISumPerClass
{
    USHORT usNumSegs; // number of segments in this class
    ULONG ulSrcWords; // sum of all source words
    ULONG ulTgtWords; // sum of all target words
} APISUMPERCLASS, *PAPISUMPERCLASS;
```

```
typedef struct _APICriteriaSum
{
    APISUMPERCLASS SimpleSum; // number of segments in this class
    APISUMPERCLASS MediumSum; // number of segments in this class
    APISUMPERCLASS ComplexSum; // number of segments in this class
} APICRITERIASUM, *PAPICRITERIASUM;
typedef struct _APIDocSaveHist
{
    APICRITERIASUM EditAutoSubst; // sums for segments translated by
    // Edit Auto
    APICRITERIASUM ExactExist; // sums for segments with exact
    // proposals
    APICRITERIASUM ExactUsed; // sums for segments with exact
    // proposals used by translator
    APICRITERIASUM FuzzyExist; // sums for segments with fuzzy
    // proposals
    APICRITERIASUM FuzzyUsed; // sums for segments with fuzzy
    // proposals used by translator
    APICRITERIASUM FuzzyExist_1; // sums for segments with fuzzy
    // proposals
    APICRITERIASUM FuzzyUsed_1; // sums for segments with fuzzy
    // proposals used by translator
    APICRITERIASUM FuzzyExist_2; // sums for segments with fuzzy
    // proposals
    APICRITERIASUM FuzzyUsed_2; // sums for segments with fuzzy
    // proposals used by translator
    APICRITERIASUM FuzzyExist_3; // sums for segments with fuzzy
    // proposals
    APICRITERIASUM FuzzyUsed_3; // sums for segments with fuzzy
    // proposals used by translator
    APICRITERIASUM MachExist; // sums for segments with machine
    // proposals
    APICRITERIASUM MachUsed; // sums for segments with machine
    // proposals used by translator
    APICRITERIASUM NoneExist; // sums for segments with no proposal
    APICRITERIASUM NotXlated; // sums for TOBE, ATTR, CURRENT
} APIDOCSAVEHIST, *PAPIDOCSAVEHIST;
```

The various classes are described in [The report layout](#). For this structure the thresholds of the standard editor were used, namely:// defines for fuzzy classes

```

#define FUZZY_THRESHOLD_1 0.7 // Threshold for different fuzzy
// classes
#define FUZZY_THRESHOLD_2 0.9 // Threshold for different fuzzy
// classes

// defines for the classes: simple sentences, medium sentences,
// complex sentences
#define SIMPLE_SENT_BOUND 5
#define MEDIUM_SENT_BOUND 15

```

Non-DDE application programming interface

OpenTM2 provides an application programming interface (API) that enables an application to directly communicate with the **OpenTM2** functions without **OpenTM2** running. However, it is required that **OpenTM2** is installed, all **OpenTM2** drives are configured, and shared resources are connected. The application can communicate with all functions currently covered by the dynamic data exchange (DDE) interface (that is, the EQFCMD command area). In addition, it can use all functions concerning dictionary and **Translation Memory** handling, namely retrieving dictionary and **Translation Memory** proposals and updating dictionaries and **Translation Memory** databases.

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 *EgfStartSession* call (see page [EgfStartSession](#)). 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 **EQFFUNC.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.

folderdefinition in non-DDE API subfolderdefinition in non-DDE API The term "folder" in the following descriptions also implies subfolders. Whenever a function requires the specification of a folder as a parameter, for example "folder_main", you can also specify a subfolder, for example "folder_2001\\folder_sub1". You can even expand subfolder specifications, up to the limits of the operating system, for example "folder_2001\\folder_sub1\\sub_sub\\sub_sub\\...".

Data types

The non-DDE interface for **OpenTM2** functions uses the following data types for parameters and return codes:

HSESSION	The session handle that is created by <i>EgfStartSession</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.
FORMLIST	A structure consisting of two length fields and a memory block. The byte <i>ch</i> indicates the start of the memory block: <i>typedef struct</i> { <i>ULONG ulAllocated;</i> <i>ULONG ulUsed;</i> <i>BYTE ch;</i> } FORMLIST, *PFORMLIST;

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 );
}
```

Calling interface reference

The following sections describe the individual calls provided by **OpenTM2**. The following calls are available:

Call...	described on page...
EqfAnalyzeDoc	...
EqfAnalyzeDocEx	...
EqfArchiveTM	...
EqfBuildSegDocName	...
EqfChangeFolProps	...
EqfChangeFolPropsEx	...
EqfChangeMFlag	...
EqfCleanMemory	...
EqfCountWords	...
EqfCreateCntReport	...
EqfCreateCountReport	...
EqfCreateControlledFolder	...
EqfCreateFolder	...
EqfCreateITM	...
EqfCreateMem	...
EqfCreateSubFolder	...

Call...	described on page...
EqfDeleteDoc	...
EqfDeleteFolder	...
EqfDeleteMem	...
EqfEndSession	...
EqfExportDoc	...
EqfExportDict	...
EqfExportFolder	...
EqfExportFolderFP	...
EqfExportFolderFPAs	...
EqfExportMem	...
EqfFreeSegFile	...
EqfGetLastError	...
EqfGetProgress	...
EqfGetSegNum	...
EqfGetSegW	...
EqfImportDoc	...
EqfImportDict	...
EqfImportFolder	...
EqfImportFolderFP	...
EqfImportMem	...
EqfLoadSegFile	...
EqfOrganizeMem	...
EqfProcessNomatch	...
EqfStartSession	...
EqfUpdateSegW	...
EqfWriteSegFile	...

EqfAnalyzeDoc

Purpose

EqfAnalyzeDoc programming interface calls **EqfAnalyzeDoc**. *EqfAnalyzeDoc* analyzes one or more documents. If no documents are specified, the function analyzes all documents in the selected folder.

This function performs the analysis in small units. Call it repetitively until it returns a return code other than CONTINUE_RC.

Format

```
usRC = EqfAnalyzeDoc(hSession,pszFolderName , ,pszDocument , ,pszMemName ,
TMMATCH_OPT ADDTOMEM_OPT AUTOSUBST_OPT UNTRANSLATED_OPT
AUTOLAST_OPT AUTOJOIN_OPT AUTOCONTEXT_OPT REDUNDCOUNT_OPT
OVERWRITE_OPT IGNOREPATH_OPT ADJUSTLEADWS_OPT ADJUSTTRAILWS_OPT
NOBLANKATSEGEND_OPT NOSUBSTIDENTICAL_OPT PROTECTXMPSCREEN_OPT
STOPATFIRSTEXACT_OPT IGNORECOMMENTED_OPT RESPECTCRLF_OPT );
```

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EqfStartSession</i> .
PSZ	pszFolderName	The name of the folder containing the documents.
PSZ	pszDocument	The name of one or more documents. If you want to analyze all documents in the folder, specify NULL or an empty list.
PSZ	pszMemName	The name of the Translation Memories to be used as search memories.
LONG	IOptions	The options to be used for the analysis:

Type	Parameter	Description
		<ul style="list-style-type: none"> • TMMATCH_OPT • ADDTOMEM_OPT • AUTOSUBST_OPT • UNTRANSLATED_OPT • AUTOLAST_OPT • AUTOJOIN_OPT • AUTOCONTEXT_OPT • REDUNDCOUNT_OPT • OVERWRITE_OPT • IGNOREPATH_OPT • ADJUSTLEADWS_OPT • ADJUSTTRAILWS_OPT • NOBLANKATSEGEND_OPT • NOSUBSTIFIDENTICAL_OPT • PROTECTXMPSCREEN_OPT • RESPECTCRLF_OPT • STOPATFIRSTEXACT_OPT • IGNORERECOMMENDED_OPT <p>These options correspond to those on the "Analyze Documents" window (see Analyzing a document). OVERWRITE_OPT must be specified if the translation of the documents has already started.</p> <p>You can combine the constants using OR.</p>

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
CONTINUE_RC	The analysis has not completed yet. Call <i>EqfAnalyzeDoc</i> again.
other	...

Code sample

```

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

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

// Analyze all documents of folder SAMPLE1 and
// substitute exact matches automatically
if ( !usRC )
{
do
{
usRC = EqfAnalyzeDoc( hSession, "SAMPLE1", NULL, ("Mem1", "Mem2"),
AUTOSUBST_OPT | OVERWRITE_OPT );
} while ( usRC == CONTINUE_RC );
} /* endif */

// terminate the session
EqfEndSession( hSession );
}

```

EwfAnalyzeDocEx

Purpose

The EwfAnalyzeDocEx programming interface calls EwfAnalyzeDocEx. EwfAnalyzeDocEx analyzes one or more documents. If no documents are specified, the function analyzes all documents in the selected folder.

This function performs the analysis in small units. Call it repetitively until it returns a return code other than CONTINUE_RC.

Format

**usRC = EwfAnalyzeDocEx(hSession,pszFolderName,
pszDocument,pszMemNames,pszProfile,pvReserved,IOptions**

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EwfStartSession</i> .
PSZ	pszFolderName	The name of the folder containing the documents.
PSZ	pszDocument	The name of one or more documents. If you want to analyze all documents in the folder, specify NULL or an empty list.
PSZ	pszMemNames	The name of one or more Translation Memories to be used as search memories. Use a comma separated list if more than one memory is specified, specify NULL if no search memory is to be used
PSZ	pszMemNames	The name of one or more Translation Memories to be used as search memories. Use a comma separated list if more than one memory is specified, specify NULL if no search memory is to be used
PSZ	pvReserved	reserved for future enhancements, specify NULL
LONG	IOptions	<p>The options to be used for the analysis:</p> <ul style="list-style-type: none"> • TMMATCH_OPT • ADDTOMEM_OPT • AUTOSUBST_OPT • UNTRANSLATED_OPT • AUTOLAST_OPT • AUTOJOIN_OPT • AUTOCONTEXT_OPT • REDUNDCOUNT_OPT • OVERWRITE_OPT • IGNOREPATH_OPT • ADJUSTLEADWS_OPT • ADJUSTTRAILWS_OPT • NOBLANKATSEGEND_OPT • NOSUBSTIFIDENTICAL_OPT • RESPECTCRLF_OPT • STOPATFIRSTEXACT_OPT • IGNORECOMMENTED_OPT <p>These options correspond to those on the "Analyze Documents" window (see Analyzing a document). OVERWRITE_OPT must be</p>

Type	Parameter	Description
		<p>specified if the translation of the documents has already started.</p> <p>You can combine the constants using OR.</p>

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
CONTINUE_RC	The analysis has not completed yet. Call <i>EqfAnalyzeDocEx</i> again.
other	Error code (EQF message number). Use <i>EqfGetLastError</i> to retrieve the complete error information.

Code sample

```

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

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

// Analyze all documents of folder SAMPLE1 and
// substitute exact matches automatically, use analysis profile Profile1
if ( !usRC )
{
do
{
usRC = EqfAnalyzeDocEX( hSession, "SAMPLE1", NULL, ("Mem1", "Mem2"), "Profile1",
NULL, AUTOSUBST_OPT | OVERWRITE_OPT );
} while ( usRC == CONTINUE_RC );
} /* endif */

// terminate the session
EqfEndSession( hSession );
}

```

ErfArchiveTM

Purpose

ErfArchiveTM programming interface calls *ErfArchiveTM*. *ErfArchiveTM* builds an Archive Translation Memory from an existing Translation Memory. At least one segment of at least one document you want to archive must have been translated (when SOURCESOURCEMEM_OPT option is not specified).

The SOURCESOURCEMEM_OPT option can be used to create a source-source Translation Memory. If the option is specified all translatable segments of the document are written to the specified Translation Memory. Without the option only segments already translated are processed.

Format

```

usRC = ErfArchiveTM(hSession, pszFolderName, '/0' chTargetDrive , NULL ,pszDocuments ,
pszMemName, OVERWRITE_OPT SOURCESOURCEMEM_OPT SETMFLAG_OPT
USEASFOLDERTM_OPT );

```

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EqfStartSession</i> .
PSZ	pszFolderName	The name of the folder.
CHAR	chTargetDrive	The target drive where the folder is located, or '/0' if it is the drive where the eqf directory is located.
PSZ	pszDocuments	List of the documents that are searched for translated

Type	Parameter	Description
		segments to be included in the Translation Memory, or NULL to search in all documents of the folder.
PSZ	pszMemName	The name of an existing Translation Memory.
LONG	IOptions	<p>The options used for the Archive Translation Memory:</p> <ul style="list-style-type: none"> • OVERWRITE_OPT (overwrites the contents of an existing Translation Memory) • USEASFOLDERTM_OPT (uses the Translation Memory as the new folder Translation Memory) • SOURCESOURCEMEM_OPT (creates a source-source Translation Memory containing all translatable segments of the document) • SETMFLAG_OPT (sets the machine translation flag of the segments written to the Translation Memory)

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
CONTINUE_RC	The Archive Translation Memory has not completed yet. Call <i>EqfArchiveTM</i> again.
other	Error code (EQF message number). Use <i>EqfGetLastError</i> to retrieve the complete error information.

Code sample

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

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

// Build Archive Translation Memory "MEM1" for the folder
// "TEST" (including document "test.txt")
if ( !usRC )
{
do
{
usRC = EqfArchiveTM(hSession, "TEST",'i',
"test.txt",
"MEM1",
OVERWRITE_OPT|USEASFOLDERTM_OPT);
} while ( usRC == CONTINUE_RC );
} //endif

// terminate the session
EqfEndSession( hSession );
}
```

EqfBuildSegDocName

Purpose

EqfBuildSegDocName programming interface calls **EqfBuildSegDocName**. Builds the fully qualified file name of a segmented document within a OpenTM2 folder.

Format

```
usRC = EqfBuildSegDocName (hSession,pszFolderName,
                           pszDocumentName,fSource,pszSegFile);
```

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EqfStartSession</i> .
PSZ	pszFolderName	Long name of the folder
PSZ	pszDocumentName	Long document name
USHORT	fSource	Flag selection source or target document <ul style="list-style-type: none"> • 0 = build segmented source file name • 1 = build segmented target file name
PSZ	pszSegFile	Points to a buffer receiving the fully qualified document file name, must have a width of at least 60 characters

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	Error code (EQF message number). Use <i>EqfGetLastError</i> to retrieve the complete error information.

```
{
USHORT usRC = 0;
CHAR szFileName [60];
HSESSION hSession = 0L;

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

if ( !usRC )
{
usRC = EqfBuildSegDocName( hSession, "SAMPLE1", "Document1", 1, szFileName );
} // endif

// terminate the session
EqfEndSession( hSession );
}
```

EqfChangeFolProps**Purpose**

EqfChangeFolProps programming interface calls *EqfChangeFolProps*. *EqfChangeFolProps* lets you change the following folder properties: the target language, the folder Translation Memory, and the dictionaries.

Format

```
usRC = EqfChangeFolProps(hSession, pszFolderName, chTargetDrive, pszTargetLanguage ,
                           pszMemName , ,pszDictionaries );
```

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EqfStartSession</i> .
PSZ	pszFolderName	The name of the folder.
CHAR	chTargetDrive	The target drive where the folder is located, if it is not the drive where the eqf directory is located. If you do not specify a drive, specify '/0'.

Type	Parameter	Description
PSZ	pszTargetLanguage	The target language for the documents in this folder, or <code>NULL</code> if the target language should not be changed. Specify the language exactly as it appears in the "Language List" window, for example <code>English(U.S.)</code> . The target language must be different from the source language.
PSZ	pszMemName	The name of the Translation Memory, or <code>NULL</code> if the Translation Memory should not be changed.
PSZ	pszDictionaries	The list of dictionaries to be used during translation. You can specify up to 10 dictionaries. If the dictionaries should not be changed, specify <code>NULL</code> .

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	Error code (EQF message number). Use <code>EqfGetLastError</code> to retrieve the complete error information.

Code sample

```

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

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

// Change the properties (target language, Memory, Dictionaries)
// of the folder named "test"
if ( !usRC )
{
usRC = EqfChangeFolProps(hSession, "test", 'e',
"English(U.S)",
"MEM1", "DICT1,DICT2");
} //endif

// terminate the session
EqfEndSession( hSession );
}

```

EqfChangeFolPropsEx

Purpose

`EqfChangeFolPropsEx` programming interface calls `EqfChangeFolPropsEx`. `EqfChangeFolPropsEx` lets you change the following folder properties: the target language, the folder Translation Memory, the dictionaries and the search Translation Memory databases.

Format

`usRC = EqfChangeFolPropsEx(hSession, pszFolderName, chTargetDrive, pszTargetLanguage ,
pszMemName , ,pszDictionaries); ,pszMemories);`

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <code>EqfStartSession</code> .
PSZ	pszFolderName	The name of the folder.

Type	Parameter	Description
CHAR	chTargetDrive	The target drive where the folder is located, if it is not the drive where the <i>Eqf</i> directory is located. If you do not specify a drive, specify '/0'.
PSZ	pszTargetLanguage	The target language for the documents in this folder, or NULL if the target language should not be changed. Specify the language exactly as it appears in the "Language List" window, for example English(U.S.). The target language must be different from the source language.
PSZ	pszMemName	The name of the Translation Memory, or NULL if the Translation Memory should not be changed.
PSZ	pszDictionaries	The list of dictionaries to be used during translation. You can specify up to 10 dictionaries. If the dictionaries should not be changed, specify NULL.
PSZ	pszMemories	The list of search Translation Memory databases to be used during analysis and translation. You can specify up to 10 Translation Memory databases. If the search Translation Memory databases should not be changed, specify NULL.

Return code
USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	Error code (EQF message number). Use <i>EqfGetLastError</i> to retrieve the complete error information.

Code sample

```

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

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

// Change the properties (target language, Memory, Dictionaries)
// of the folder named "test"
if ( !usRC )
{
    usRC = EqfChangeFolProps(hSession, "test", 'e',
    "English(U.S)",
    "MEM1", "DICT1,DICT2");
} //endif

// terminate the session
EqfEndSession( hSession );
}

```

EfqChangeMFlag

Purpose

EfqChangeMFlag programming interface calls **EfqChangeMFlag**. Segments that were translated by machine are prefixed with an [m]. OpenTM2 provides a command to have these m prefixes removed from machine-translated segments in a Translation Memory. Alternatively, this function lets you add m flags to segments that did not have such a flag before.

Format

```
usRC = EqfChangeMFlag(hSession, pszTransMem, ); IAction SET_MMOPt CLEAR_MMOPt
```

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EfqStartSession</i> .
PSZ	pszTransMem	The name of the Translation Memory that you want to work with.
LONG	IAction	Specifies whether you want to remove (CLEAR_MMOPt) or set (SET_MMOPt) the m flags in the specified Translation Memory.

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	Error code (EQF message number). Use <i>EfqGetLastError</i> to retrieve the complete error information.

Code sample

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

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

// Remove m flags in Translation Memory TestTM.
if ( !usRC )
{
usRC = EqfCreateTM(hSession, "TestTM", CLEAR_MMOPt);
} //endif

// terminate the session
EqfEndSession( hSession );
}
```

EfqCleanMemory

Purpose

EfqCleanMemory programming interface calls **EfqCleanMemory**. The API call *EfqCleanMemory* removes all segments which are not relevant for a given translation package from an external memory. The “cleaned” memory can be created in internal or external format.

This function performs the cleanup in small units. Call it repetitively until it returns a return code other than CONTINUE_RC.

Format

```
usRC = EqfCleanMemory(hSession,pszFolder,pszInMem, pszOutMem,IOptions);
```

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EfqStartSession</i> .
PSZ	pszFolder	The name of a OpenTM2 folder (already imported into

Type	Parameter	Description
		TM and the documents have to be analyzed).
PSZ	pszInMem	The fully qualified file name of the input memory in Ansi or UTF-16 encoding.
PSZ	pszOutMem	The name of an new internal memory or the fully qualified name of an external.
LONG	IOptions	<p>The option to be used for the cleanup of a memory:</p> <ul style="list-style-type: none"> • CLEANMEM_INTERNAL_MEMORY_OPT to create an internal memory • CLEANMEM_EXTERNAL_MEMORY_OPT to create an external memory (default) • OVERWRITE_OPT to overwrite any existing output memory • CLEANMEM_COMPLETE_IN_ONE_CALL_OPT If set the API call does not return after each processing step but stays in the API call until the function has been completed • CLEANMEM_BESTMATCH_OPT if set only the best match is written to the output memory, if not set the best three matches are written to the output memory • CLEANMEM_MERGE_OPT when specified the cleaned memory matches are merged into an existing memory rather than creating a new one • CLEANMEM_KEEP_DUPS_OPT when specified duplicate exact matches are left in the memory (without this option only the first exact match is left in the memory). Fuzzy matches are left in the memory as long there is no exact match for the same segment (without this option only the best fuzzy match is left in the memory)

Return code
USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
CONTINUE_RC	The memory cleanup has not been completed yet. Call again.
other	Error code (EQF message number). Use ... to retrieve the complete error information.

Code sample
HSESSION hSession;

```

USHORT usRC;

usRC = EqfStartSession( &hSession );
usRC = EqfCleanMemory( "TestFolder",
"C:\EXPMEMORY\SAMPLE2.EXP",
"C:\EXPMEMORY\SAMPLEOUT.EXP",
CLEANMEM_EXTERNAL_MEMORY_OPT | CLEANMEM_COMPLETE_IN_ONE_CALL_OPT |
OVERWRITE_OPT );

usRC = EqfEndSession( hSession );

```

ErfCountWords

Purpose

ErfCountWords programming interface calls **ErfCountWords**. *ErfCountWords* counts the words of one or more documents. This function performs the counting in small units. Call it repetitively until it returns a return code other than CONTINUE_RC.

Format

```
usRC = ErfCountWords(hSession,pszFolderName, pszDocuments,IOptions,pszOutFile);
```

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>ErfStartSession</i> .
PSZ	pszFolderName	The name of the folder containing the documents which are to be counted.
PSZ	pszDocuments	The pointer to a list of documents or NULL if no documents are specified. If no documents are specified, the words of all documents in the folder are counted.
LONG	IOptions	<p>The options to be used for the counting:</p> <ul style="list-style-type: none"> • SOURCE_OPT (source word count) • TARGET_OPT (translated/untranslated word count) • TMMATCH_OPT (memory match count) • DUPLICATE_OPT (count duplicate words) • DUPMEMMATCH_OPT (count duplicate words and include memory match information) <p>For the TMMATCH_OPT the following option can be specified</p> <ul style="list-style-type: none"> • SEPERATEREPLMATCH_OPT to count replace matches seperately <p>These constants are mutually exclusive, they can be combined with the format of the output file:</p> <ul style="list-style-type: none"> • XML_OUTPUT_OPT (output as XML file) or • TEXT_OUTPUT_OPT (output in text format) or • HTML_OUTPUT_OPT (output in HTML format) and the OVERWRITE_OPT (to overwrite existing output files) using " " (bitwise OR)

Type	Parameter	Description
		operator). If no output format is specified TEXT_OUTPUT_OPT is used as default.
PSZ	pszOutFile	The fully qualified name of the output file. If the file already exists, specify the OVERWRITE_OPT option (otherwise this call fails).

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
CONTINUE_RC	Word counting has not completed yet. Call <i>EqfCountWords</i> again.
other	

Code sample

```

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

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

// Count the source (=original) words of all documents in folder
// SAMPLE1 and store the counting results in file C:\COUNT.OUT
if ( !usRC )
{
do
{
usRC = EqfCountWords( hSession, "SAMPLE1", NULL, SOURCE_OPT,
"C:\\COUNT.OUT" );
} while ( usRC == CONTINUE_RC );
} /* endif */

// terminate the session
EqfEndSession( hSession );
}

```

EqfCreateCntReport

Purpose

EqfCreateCntReport programming interface calls **EqfCreateCntReport**. *EqfCreateCntReport* creates Calculating, Preanalysis, Redundancy, Redundant segment list reports.

Format

usRC = EqfCreateCntReport(hSession, pszFolderName, ,pszDocuments , ,pszOutfileName, pszFormat, pszProfile, , [usColumn] [usCategory], usColumn, usCategory, , PLAUS_OPT LOST_OPT LIST_OPT , bSingleShipment); structReportType pszReport, IRepType, pszDescription structRepSettings pszCountType, bShow, bSummary, pszRepLayout, bShrink, pszStatisticType, bExProposal structFactSheet IComplexity, IPayfactor structFinalFactors pszUnit, ICurrFactor, pszLocalCurrency

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EqfStartSession</i> .
PSZ	pszFolderName	The name of the folder.
PSZ	pszDocuments	List of documents, or NULL if all documents of the folder should be used.
PREPORT TYPE	pstructReportType	See ...for details.
PSZ	pszOutfileName	The name of the file where the report is to be stored

Type	Parameter	Description
		(along with the drive and directory information).
PSZ	pszFormat	Format of the Output file ("ASCII", "HTML", or "XML").
PSZ	pszProfile	The name of the profile to be loaded, or NULL.
PREPORT SETTINGS	pstructRepSettings	See ... for details.
PFACTSHEET	pstructFactSheet [usColumn] [usCategory]	Array of See ... for details.
USHORT	usColumn	The first array index represents the column number according to the listed columns in the dialog "Create Counting Report", tab "Fact Sheet".
USHORT	usCategory	The second array index represents the category number according to the listed categories in the dialog "Create Counting Report", tab "Fact Sheet".
PFINAL FACTORS	pstructFinalFactors	See ... for details.
LONG	IOptSecurity	The options to be used for security: <ul style="list-style-type: none"> • PLAUS_OPT (Plausibility check) • LOST_OPT (Lost Data: Force new shipment) • LIST_OPT (List of Documents) You can combine the options using OR.
BOOL	bSingleShipment	<ul style="list-style-type: none"> • TRUE = Single Shipments • FALSE = All Shipments

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	Error code (EQF message number). Use <i>EqfGetLastError</i> to retrieve the complete error information.

Code sample

```
{
USHORT usRC = 0;
HSESSION hSession = 0L;
int i,j;
#define COLUMN 10;
#define CATGORY 3;

REPORTTYPE ReportType = {NULL, 0L, NULL};
REPSETTINGS ReportSettings = {NULL, 0, 0, NULL, 0, NULL, 0};
FACTSHEET FactSheet[COLUMN][CATEGORY];
FINALFACT FinalFactors = {0L, 0L NULL};

//fill ReportType structure
ReportType.pszReport = "Calculating Report";
ReportType.lRepType=BASE_TYP | FACT_TYP | SUM_TYP;
ReportType.pszDescription[0]='\0';

//fill ReportSettings strucure
RepSettings.pszCountType = "Source Words";
RepSettings.bShow=TRUE;
RepSettings.bSummary=TRUE;
```

```

RepSettings.pszRepLayout = "Standard";
RepSettings.bShrink=FALSE;
RepSettings.pszStatisticType = NULL;
RepSettings.bExProposal=FALSE;

//fill FactSheet structure
for(i=0;i++,i<COLUMN)
{
for(j=0,j++,j<CATEGORY)
{
FactSheet[i][j].lComplexity = (float)1.0;
FactSheet[i][j].lPayFactor = (float)1.0;
}
}

// fill FinalFactors structure
FinalFactors.lUnit = 1;
FinalFactors.lCurrFactor = (float)1.0;
FinalFactors.pszLocalCurrency = "EUR";

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

if ( !usRC )
{
usRC = EqfCreateCntReport(hSession, 'e', "TEST", "test.doc",
test2.doc", &ReportType,
"E:\\Project\\CalcReport", "HTML",
NULL,
&RepSettings,(void *)FactSheet,
COLUMN, CATEGORY, &FinalFactors,
PLAUS_OPT, TRUE);
} //endif
// terminate the session
EqfEndSession( hSession );
}

```

Parameters for structReportType

```

typedef struct _REPORTTYPE
{
    PSZ pszReport;
    LONG lRepType;
    PSZ pszDescription;
} REPORTTYPE, *PREPORTTYPE;

```

Type	Parameter	Description
PSZ	pszReport	Specifies one of the following reports: <ul style="list-style-type: none"> • "Calculating Report" • "Pre-Analysis Report" • "Redundancy Report" • "Redundant Segment List"
LONG	lRepType	One, or a combination, of the following report types: <ul style="list-style-type: none"> • BASE_TYP • FACT_TYP • SUM_TYP Allowed combinations are: <ul style="list-style-type: none"> • Base • Summary • Fact Sheet • Base & Summary • Summary & Fact Sheet • Base, Summary & Fact Sheet
PSZ	pszDescription	The report description, or NULL.

Parameters for structRepSettings

```
typedef struct _REPORTSETTINGS
{
    PSZ pszCountType;
    BOOL bShow;
    BOOL bSummary;
    PSZ pszRepLayout;
    BOOL bShrink;
    PSZ pszStatisticType;
    BOOL bExProposal;
} REPORTSETTINGS, *PREPORTSETTINGS;
```

Type	Parameter	Description
PSZ	pszCountType	Specifies what to count: • "Source Words" • "Target Words" • "Segments" • "Modified Words"
BOOL	bShow	• TRUE = Show categories • FALSE = Hide categories
BOOL	bSummary	Build summary of categories
PSZ	pszRepLayout	Specify one of the following layouts: • "Standard" • "Standard and Group-Summary" • "Shrunk to Groups"
BOOL	bShrink	Automatic Shrink
PSZ	pszStatisticType	For pszReport = "Calculating Report" specify one of the following keywords: • "Standard" • "Advanced" or NULL for all other reports or no statistics.
BOOL	bExProposal	Use Existing Proposals.

Parameters for structFactSheet

```
typedef struct _FACTSHEET
{
    LONG lComplexity;
    LONG lPayFactor;
} FACTSHEET, *PFACTSHEET;
```

Type	Parameter	Description
float	lComplexity	Specifies the Complexity Factor.
float	lPayFactor	Specifies the Pay Factor.

Parameters for structFinalFactors

```
typedef struct _FINALFACTORS
{
    LONG lUnit;
    LONG lCurrFactor;
    PSZ pszLocalCurrency;
} FINALFACTORS, *PFINALFACTORS;
```

Type	Parameter	Description
LONG	lUnit	Values (in words): • 1 • 10 • 250
float	lCurrFactor	Specifies the local currency factor.
PSZ	pszLocalCurrency	Specifies the local currency. The local currencies correspond to the values of the dialog "Create Counting Report", tab "Fact Sheet".

ErfCreateCountReport

Purpose

The ErfCreateCountReport programming interface calls ErfCreateCountReport. ErfCreateCountReport creates Calculating, Preanalysis, Redundancy, Redundant segment list reports using the supplied profile.

Format

```
usRC = ErfCreateCountReport(hSession, pszFolderName,pszDocuments,  
    pszOutfileName,useReport usType, pszProfile,IOptions);
```

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by ErfStartSession.
PSZ	pszFolderName	The name of the folder.
PSZ	pszDocuments	List of documents, or NULL if all documents of the folder should be used.
PSZ	pszOutfileName	The name of the file where the report is to be stored (along with the drive and directory information).
PSZ	pszProfile	The name of the profile to be loaded.
USHORT	usReport	Type of report: <ul style="list-style-type: none">• HISTORY REP (History Report)• COUNTING REP (Counting Report)• CALCULATING REP (Calculation Report)• PREANALYSIS REP (PreAnalysis Report)• REDUNDANCY REP (Redundancy Report)• REDUNDANCYSEGMENT REP (Redundancy Segment List)
USHORT	usType	Type of report: for HISTORY REP <ul style="list-style-type: none">• BRIEF_SORTBYDATE_REPORTTYPE• BRIEF_SORTBYDOC_REPORTTYPE• DETAIL_REPORTTYPE for HISTORY REP <ul style="list-style-type: none">• WITHTOTALS_REPORTTYPE• WITHOUTTOTALS_REPORTTYPE for CALCULATING REP, PREANALYSIS REP, and REDUNDANCY REP <ul style="list-style-type: none">• BASE_REPORTTYPE• BASE_SUMMARY_REPORTTYPE• BASE_SUMMARY_FACTSHEET_REPORTTYPE• SUMMARY_FACTSHEET_REPORTTYPE• FACTSHEET_REPORTTYPE
LONG	IOptions	Options for the counting report: OVERWRITE_OPT or 0

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	Error code (EQF message number). Use <i>EqfGetLastError</i> to retrieve the complete error information.

Code sample

```
{
USHORT usRC = 0;
HSESSION hSession = 0L;
// start the Eqf calling interface session
usRC = EqfStartSession(&hSession);

if ( !usRC )
{
    usRC = EqfCreateCountReport(hSession, 'e', "TEST", "test.doc,
    test2.doc",
    "E:\\Project\\CalcReport",
    COUNTING_REPORT, BASESUMMARY_REPTYPE
    "PUB0205", OVERWRITE_OPT);
} //endif
// terminate the session
EqfEndSession( hSession );
}
```

EqfCreateControlledFolder

Purpose

EqfCreateControlledFolder programming interface calls **EqfCreateControlledFolder**. *EqfCreateControlledFolder* creates a new controlled folder by using the specified values. Configure the target drive for the folder using the "Configure Drives" window of OpenTM2.

Format

```
usRC = EqfCreateControlledFolder(hSession,pszFolderName, pszDescription , chTargetDrive ,
pszTransMem, pszMarkup, pszEditor , .pszDictionaries , pszSourceLanguage,
pszTargetLanguage, pszConversion , .pszReadOnlyMems , pszPassword, pszProjCoordName ,
pszProjCoordMail , pszTranslatorName , pszTranslatorMail , pszProductName , pszProductFamily,
pszSimilarProduct , pszProductDict , pszProductMem , pszPreviousVersion , pszVersion ,
pszShipmentNumber );
```

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EqfStartSession</i> .
PSZ	pszFolderName	The name of the folder to be created.
PSZ	pszDescription	The folder description, or NULL.
CHAR	chTargetDrive	The target drive for the new folder. If omitted, the primary EQF drive is used. The drive must be the primary EQF drive or one of the secondary EQF drives defined in the "Configure Drives" window.
PSZ	pszTransMem	The name of the Translation Memory to be used for the documents in the new folder.
PSZ	pszMarkup	The name of the markup table, for example EQFMRI.
PSZ	pszEditor	The name of the editor. If not specified, the editor STANDARD is used.
PSZ	pszDictionaries	The list of dictionaries to be used during translation. You can specify up to 10 dictionaries.
PSZ	pszSourceLanguage	The source language for the

Type	Parameter	Description
		documents in this folder. Specify the language exactly as it appears in the "Language List" window, for example English (U.S.).
PSZ	pszTargetLanguage	The target language for the documents in this folder. Specify the language exactly as it appears in the "Language List" window, for example English (U.S.). The target language must be different from the source language.
PSZ	pszConversion	The export conversion type, or NULL for no conversion.
PSZ	pszReadOnlyMems	The list of Translation Memories to search through during translation, or NULL. You can specify up to 4 Translation Memories.
PSZ	pszPassword	The password to protect the folder against changes. The password can be up to six characters long.
PSZ	pszProjCoordName	The name of the project coordinator, or NULL.
PSZ	pszProjCoordMail	The e-mail address of the project coordinator, or NULL.
PSZ	pszTranslatorName	The name of the translator responsible for this folder, or NULL.
PSZ	pszTranslatorMail	The e-mail address of the translator, or NULL.
PSZ	pszProductName	The product name this folder is assigned to, or NULL.
PSZ	pszProductFamily	The product family this folder is assigned to, or NULL.
PSZ	pszSimilarProduct	The name of a similar product this folder is assigned to, or NULL.
PSZ	pszProductDict	The product-specific dictionary to be used during translation, or NULL.
PSZ	pszProductMem	The product-specific memory to be used during translation, or NULL.
PSZ	pszPreviousVersion	The previous version number of the product specified above, or NULL.
PSZ	pszVersion	The actual version number of the product specified above, or NULL.
PSZ	pszShipmentNumber	The number of the shipment, or NULL.

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	Error code (EQF message number). Use <i>EqfGetLastError</i> to retrieve the complete error information.

Code sample

```

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

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

// Create a new controlled folder name 'Test' on the
// primary Eqf drive
if ( !usRC )
{
usRC = EqfCreateControlledFolder(hSession, "Test",
"Description of folder Test",
'\0', // use primary Eqf drive
"MEM1", "EQFASCII",
"STANDARD", "DICT1,ENGLGERM",
"English(U.S.)","German(national)",
NULL, NULL, "passwd",
"ProjCoordName", "ProjCoordMail",
"TranslatorName","TranslatorMail", NULL,
"Family", NULL, "Dict", "MemoryName",
"1.0", "2.0", "1");
} //endif

// terminate the session
EqfEndSession( hSession );
}

```

EqfCreateFolder**Purpose**

EqfCreateFolder programming interface calls *EqfCreateFolder*. *EqfCreateFolder* creates a new folder by using the specified values. Configure the target drive for the folder using the "Configure Drives" window of OpenTM2.

Format

```

usRC = EqfCreateFolder(hSession,pszFolderName,
pszDescription,chTargetDrive,pszTransMem,pszMarkup,
pszEditor ,pszDictionaries ,pszSourceLanguage,
pszTargetLanguage,pszConversion,pszROMemory);

```

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EqfStartSession</i> .
PSZ	pszFolderName	The name of the folder to be created.
PSZ	pszDescription	The folder description, or NULL.
CHAR	chTargetDrive	The target drive for the new folder. If omitted, the primary EQF drive is used. The drive must be the primary EQF drive or one of the secondary EQF drives defined in the "Configure Drives" window.
PSZ	pszTransMem	The name of the Translation Memory to be used for the documents

Type	Parameter	Description
		in the new folder.
PSZ	pszMarkup	The name of the markup table, for example EQMRI.
PSZ	pszEditor	The name of the editor. If not specified, the editor STANDARD is used.
PSZ	pszDictionaries	The list of dictionaries to be used during translation. You can specify up to 10 dictionaries.
PSZ	pszSourceLanguage	The source language for the documents in this folder. Specify the language exactly as it appears in the "Language List" window, for example English (U.S.).
PSZ	pszTargetLanguage	The target language for the documents in this folder. Specify the language exactly as it appears in the "Language List" window, for example English (U.S.).
PSZ	pszConversion	Conversion to be used for the folder or NULL
PSZ	pszROMemory	List of read-only memories to be searched or NULL

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	...

Code sample

```

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

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

// Create a new folder name SAMPLE1 on the primary Eqf drive
if ( !usRC )
{
    usRC = EqfCreateFolder( hSession, "SAMPLE1",
    "Description of folder SAMPLE1",
    '\0', // use primary Eqf drive
    "MEM01", "EQFASCII", "STANDARD",
    "DICT1,ENGLGERM", "English(U.S.)",
    "German(National)" );

} /* endif */

// terminate the session
EqfEndSession( hSession );
}

```

EqfCreateITM

Purpose

EqfCreateITM programming interface calls **EqfCreateITM**. **EqfCreateITM** creates an Initial Translation Memory (ITM) database from an existing Translation Memory. It can create an internal Translation Memory and an external Translation Memory. The internal Translation Memory must not be filled.

Important hint: If you want to generate a source English-English memory (i.e. a memory where the source sentence and the target sentence are identical), please always use EQFArchiveTM function

with the option SOURCE_SOURCEMEM_OPT.

Format

```
usRC = EqfCreateITM(hSession, pszMemDB, pszMarkup, pszSGMLMemFile,
pszSourceLanguage, pszTargetLanguage, pszSourceStartPath, pszTargetStartPath, );
pszFilePairs ,pszSourceFile,pszTranslationFile , IType|NOANA_TYP NOTM_TYP PREPARE_TYP
```

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EwfStartSession</i> .
PSZ	pszMemDB	The name of a previously created OpenTM2 Translation Memory (without the file extension). This Translation Memory can still be empty. It can be filled with original segments and their corresponding translations.
PSZ	pszFilePairs	List of file names to use when creating the ITM, in the form (original1, translation1, original2, translation2).
PSZ	pszMarkup	The name of the markup table, for example EQFMRI.
PSZ	pszSGMLMemFile	The name you want to give to the external ITM, and the path where it is to be located. The ITM is in SGML format and can subsequently be imported into OpenTM2 after you have checked it.
PSZ	pszSourceLanguage	The source language for the documents in this folder. Specify the language exactly as it appears in the "Language List" window, for example English (U.S.).
PSZ	pszTargetLanguage	The target language for the documents in this folder. Specify the language exactly as it appears in the "Language List" window, for example English (U.S.).
PSZ	pszSourceStartPath	<p>The path information that you do <i>not</i> want to become part of the document name when the original document is stored in the Initial Translation Memory.</p> <p>For example, if your source file is stored in e:\tm\project\english, and you do not want e:\tm\project to become part of the name under which it is stored, specify e:\tm\project.</p> <p>The path you specify here can differ from the <i>pszTargetStartPath</i>. However, if you specify a source start path, you must</p>

Type	Parameter	Description
		<p>also specify a <i>pszTargetStartPath</i>.</p>
PSZ	pszTargetStartPath	<p>The path information that you do want to become part of the document name when the target document is stored in the Initial Translation Memory.</p> <p>For example, if your source file is stored in e:\tm\project\english and you do not want e:\tm\project to become part of the name under which it is stored, specify e:\tm\project.</p> <p>The path you specify here can differ from the <i>pszSourceStartPath</i>. However, if you specify a source start path, you must also specify a <i>pszSourceStartPath</i>.</p>
LONG	IType	<p>One or more of the following:</p> <ul style="list-style-type: none"> • NOANA_TYP Do not analyze the selected files because they have already been analyzed by OpenTM2. • NOTM_TYP Do not fill the internal Translation Memory (<i>pszMemDB</i>). Fill the external Translation Memory. It is in SGML format and you can check it afterwards. • PREPARE_TYP The source documents are related to their corresponding translations. The file pairs are prefixed with p. You can combine the options using OR.

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	Error code (EQF message number). Use <i>EqfGetLastError</i> to retrieve the complete error information.

Code sample

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

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

// Create a new Initial Translation Memory TestITM
if ( !usRC )
{
usRC = EqfCreateITM(hSession, "TestITM",
"E:\TM\PROJECT\ENGLISH\original1",
"E:\TM\PROJECT\GERMAN\translation1",
"EQFASCII", NULL,
```

```

"English(U.S.)", "German(national)",
"E:\TM\PROJECT", "E:\TM\PROJECT", 0);
} //endif

// terminate the session
EqfEndSession( hSession );
}

```

EqfCreateMem

Purpose

EqfCreateMem programming interface calls **EqfCreateMem**. *EqfCreateMem* creates a new shared or local **Translation Memory**.

Format

```

usRC = EqfCreateMem(hSession,pszMemName
,pszDescription,chToDrive,pszSourceLanguage,IOptions);

```

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EqfStartSession</i> .
PSZ	pszMemName	The name of the Translation Memory to be created.
PSZ	pszDescription	The description of the Translation Memory .
CHAR	chToDrive	The target drive for the new Translation Memory . If omitted, the primary EQF drive is used. The drive must be the primary EQF drive or one of the secondary EQF drives defined in the "Configure Drives" window.
LONG	IOptions	The type of the new Translation Memory : <ul style="list-style-type: none"> • LOCAL_OPT, which is the default • SHARED_OPT
PSZ	pszSourceLanguage	The source language to be used for the Translation Memory

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	...

Code sample

```

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

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

// Create the new local Translation Memory MEMDB2 on the
// primary Eqf system drive
if ( !usRC )
{
usRC = EqfCreateMem( hSession, "MEMDB2",
"TM created via Func I/F",
"\0", "English(U.S.)", LOCAL_OPT );
} /* endif */

// terminate the session
EqfEndSession( hSession );

```

}

EwfCreateSubFolder

Purpose

EwfCreateSubFolder programming interface calls EwfCreateSubFolder. EwfCreateSubFolder creates a subfolder from a parent folder by using the specified values. The parent folder itself can be a subfolder.

Format

```
usRC = EwfCreateSubFolder(hSession, pszParentFolName, pszSubFolName, pszMemName,  
    pszMarkup, pszSourceLanguage, pszTargetLanguage, pszEditor, pszConversion, pszTranslator,  
    pszTranslatorMail);
```

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by EwfStartSession.
PSZ	pszParentFolName	The name of the parent folder, or the name of a subfolder that acts as a parent folder.
PSZ	pszSubFolName	The name of the subfolder to be created.
PSZ	pszMemName	The name of the Translation Memory to be used for the documents in the new folder. If you want the same as in the parent folder, specify NULL.
PSZ	pszMarkup	The name of the markup table, for example EQFMRI. If you want the same as in the parent folder, specify NULL.
PSZ	pszSourceLanguage	The source language for the documents in this folder. Specify the language exactly as it appears in the "Language List" window, for example English(U.S.). If you want the same as in the parent folder, specify NULL.
PSZ	pszTargetLanguage	The target language for the documents in this folder. Specify the language exactly as it appears in the "Language List" window, for example English(U.S.). The target language must differ from the source language. If you want the same as in the parent folder, specify NULL.
PSZ	pszEditor	The name of the editor. If not specified, the editor STANDARD is used.
PSZ	pszConversion	The export conversion type. If you want the same as in the parent folder, specify NULL.
PSZ	pszTranslator	The name of the translator. If you want the same as in the parent folder, specify NULL.
PSZ	pszTranslatorMail	The e-mail address of the translator. If you want the same as in the parent folder, specify NULL.

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	Error code (EQF message number). Use <i>EqfGetLastError</i> to retrieve the complete error information.

Code sample

```
{  
USHORT usRC = 0;  
HSESSION hSession = 0L;  
  
// start the Eqf calling interface session  
usRC = EqfStartSession(&hSession);  
  
// Create a subfolder "SUBSUBTEST" of the parent folder "SUBTEST",  
// which itself is a subfolder of parent folder "TEST".  
  
if ( !usRC )  
{  
usRC = EqfCreateSubFolder(hSession,  
"TEST\\SUBTEST", "SUBSUBTEST",  
"MEM1", "EQFASCII",  
"English(U.S.)",  
"German(national)", NULL, NULL,  
"Translator",  
"Translator@xyz.com");  
} //endif  
  
// terminate the session  
EqfEndSession( hSession );  
}
```

EqfDeleteDoc

Purpose

EqfDeleteDoc programming interface calls *EqfDeleteDoc*. *EqfDeleteDoc* deletes the specified documents.

Format

usRC = EqfDeleteDoc(hSession,pszFolderName, pszDocuments);

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EqfStartSession</i> .
PSZ	pszFolderName	The name of the folder containing the documents to be deleted.
PSZ	pszDocuments	The name of the documents to be deleted.

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	...

Code sample

```
{  
USHORT usRC = 0;  
HSESSION hSession = 0L;  
  
// start the Eqf calling interface session  
usRC = EqfStartSession( &hSession );  
  
// Delete document DOC1.TXT in folder SAMPLE1
```

```

if ( !usRC )
{
usRC = EqfDeleteDoc( hSession, "SAMPLE1", "DOC1.TXT" );
} /* endif */

// terminate the session
EqfEndSession( hSession );
}

```

EqfDeleteFolder

Purpose

EqfDeleteFolder programming interface calls**EqfDeleteFolder** *EqfDeleteFolder* deletes the specified folder and all the documents that it contains.

Format

```
usRC = EqfDeleteFolder(hSession,pszFolderName);
```

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EqfStartSession</i> .
PSZ	pszFolderName	The name of the folder to be deleted.

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	...

Code sample

```

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

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

// Delete the folder SAMPLE1
if ( !usRC )
{
usRC = EqfDeleteFolder( hSession, "SAMPLE1" );
} /* endif */

// terminate the session
EqfEndSession( hSession );
}

```

EqfDeleteMem

Purpose

EqfDeleteMem programming interface calls**EqfDeleteMem** *EqfDeleteMem* deletes a **Translation Memory**.

Format

```
usRC = EqfDeleteMem(hSession,pszMemName);
```

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EqfStartSession</i> .
PSZ	pszMemName	The name of the Translation Memory to be deleted.

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	...

Code sample

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

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

// Delete the Translation Memory MEMDB2
if ( !usRC )
{
    usRC = EqfDeleteMem( hSession, "MEMDB2" );
} /* endif */

// terminate the session
EqfEndSession( hSession );
}
```

EqfEndSession

Purpose

EqfEndSession programming interface calls EqfEndSession. EqfEndSession terminates an EQF batch session and cleans up all resources used by the batch function.

Format

usRC = EqfEndSession(hSession);

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by EqfStartSession .

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	Error code (EQF message number). You cannot use EqfGetLastError to retrieve complete error information if a call to EqfEndSession failed.

EqfExportDict

Purpose

EqfExportDict programming interface calls EqfExportDict. EqfExportDict exports a dictionary in SGML format to the specified file. It fails if the output file exists already unless the OVERWRITE_OPT has been set. Default encoding of output SGML dictionary is Unicode (UTF16). Specify the option ASCII_OPT or ANSI_OPT if the export dictionary should have the corresponding format.

This function performs the export in small units. Call it repetitively until it returns a return code other than CONTINUE_RC.

Format

usRC = EqfExportDict(hSession,pszDictName, OVERWRITE_OPT ASCII_OPT ANSI_OPT UTF16_OPT , pszOutFile);

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by EqfStartSession.
PSZ	pszDictName	The name of the dictionary to be exported.

Type	Parameter	Description
LONG	IOptions	The option to be used for the export: <ul style="list-style-type: none"> • OVERWRITE_OPT • ASCII_OPT • ANSI_OPT • UTF16_OPT
PSZ	pszOutFile	The fully qualified name of the output file. If the output file exists already, specify the OVERWRITE_OPT option.

Return code
USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
CONTINUE_RC	The dictionary export has not completed yet. Call <i>EqfExportDict</i> again.
other	...

Code sample

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

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

// Export the dictionary ENGLGERM to SGML file C:\DICT1.SGM
// and overwrite any existing SGML file with this name
if ( !usRC )
{
do
{
usRC = EqfExportDict( hSession, "ENGLGERM", OVERWRITE_OPT | UTF16_OPT
"C:\\DICT1.SGM" );
} while ( usRC == CONTINUE_RC );
} /* endif */

// terminate the session
EqfEndSession( hSession );
}
```

EqfExportDoc

Purpose

EqfExportDoc programming interface calls *EqfExportDoc*. *EqfExportDoc* exports one or more documents.

This function performs the export in small units. Call it repetitively until it returns a return code other than CONTINUE_RC.

Format

`usRC = EqfExportDoc(hSession,pszFolderName,pszFiles,pszStartPath ,pszFiles ,
TARGET_OPT SOURCE_OPT SNOMATCH_OPT OVERWRITE_OPT);`

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EqfStartSession</i> .
PSZ	pszFolderName	The name of the folder containing the documents to be exported.
PSZ	pszStartPath	The start path if the documents are to be exported with relative path information. If a start path is specified, the files in the list only contain the relative path.

Type	Parameter	Description
PSZ	pszFiles	The name, including the target path, of the documents to be exported.
LONG	IOptions	<p>The options to be used for the document export:</p> <ul style="list-style-type: none"> • TARGET_OPT, which is the default • SOURCE_OPT • SNOMATCH_OPT • OVERWRITE_OPT to replace existing documents <p>These options correspond to those in the "Export Documents" window (see Exporting a document in external format).</p>

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
CONTINUE_RC	The document export has not completed yet. Call <i>EqfExportDoc</i> again.
other	...

Code sample

```

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

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

// Export the translations of documents DOC1.TXT and DOC2.TXT of
// folder SAMPLE1
if ( !usRC )
{
do
{
usRC = EqfExportDoc( hSession, "SAMPLE1", NULL,
"C:\\DOC1.TXT,C:\\DOC2.TXT",
TARGET_OPT );

} while ( usRC == CONTINUE_RC );
/* endif */

// terminate the session
EqfEndSession( hSession );
}

```

EqfExportFolder

Purpose

EqfExportFolder programming interface calls *EqfExportFolder*. *EqfExportFolder* exports a folder to a specific target drive. The path is always \eqf\export.

This function performs the export in small units. Call it repetitively until it returns a return code other than CONTINUE_RC.

Format

*usRC = EqfExportFolder(hSession,pszFolderName, chTargetDrive , WITHMEM_OPT
WITHREADONLYMEM_OPT WITHDOCMEM_OPT DELETE_OPT WITHDICT_OPT
MASTERFOLDER_OPT XLIFF_OPT OVERWRITE_OPT ,pszDocuments ,pszDescription);*

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EwfStartSession</i> .
PSZ	pszFolderName	The name of the folder to be exported.
PSZ	pszDescription	The folder description, or NULL.
CHAR	chTargetDrive	The drive to which the folder is exported.
LONG	IOptions	<p>The options to be used for the export:</p> <ul style="list-style-type: none"> • WITHMEM_OPT • WITHREADONLYMEM_OPT • WITHDOCMEM_OPT • DELETE_OPT • WITHDICT_OPT • MASTERFOLDER_OPT • XLIFF_OPT • OVERWRITE_OPT <p>These options correspond to those in the "Export Folder" window (see Exporting a folder).</p> <p>You can combine the constants using OR.</p>
PSZ	pszDocuments	The name of one or more documents.

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
CONTINUE_RC	The folder export has not completed yet. Call <i>EwfExportFolder</i> again.
other	...

Code sample

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

// start the Ewf calling interface session
usRC = EwfStartSession( &hSession );

// Export the folder SAMPLE1 to drive C: with the folder
// Translation Memory and all folder dictionaries, overwrite
// any previously exported folder on drive C:
if ( !usRC )
{
do
{
usRC = EwfExportFolder( hSession, "SAMPLE1", 'C',
WITHMEM_OPT | WITHDICT_OPT | OVERWRITE_OPT,
NULL, NULL );
} while ( usRC == CONTINUE_RC );
} /* endif */

// terminate the session
EwfEndSession( hSession );
}
```

EwfExportFolderFP

Purpose

EwfExportFolderFP programming interface calls *EwfExportFolderFP*. *EwfExportFolderFP* exports a folder to a specific path.

This function performs the export in small units. Call it repetitively until it returns a return code other than CONTINUE_RC.

Format

```
usRC = EqfExportFolderFP(hSession,pszFolderName,pszTargetPath , WITHMEM_OPT  
WITHREADONLYMEM_OPT WITHDOCMEM_OPT DELETE_OPT WITHDICT_OPT  
MASTERFOLDER_OPT XLIFF_OPT OVERWRITE_OPT ,,,pszDocuments ,pszDescription);
```

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EwfStartSession</i> .
PSZ	pszFolderName	The name of the folder to be exported.
PSZ	pszDescription	The folder description, or NULL.
PSZ	pszTargetPath	The path to which the folder is exported.
LONG	IOptions	<p>The options to be used for the export:</p> <ul style="list-style-type: none"> • WITHMEM_OPT • WITHREADONLYMEM_OPT • WITHDOCMEM_OPT • DELETE_OPT • WITHDICT_OPT • MASTERFOLDER_OPT • XLIFF_OPT • OVERWRITE_OPT <p>These options correspond to those in the "Export Folder" window (see Exporting a folder).</p> <p>You can combine the constants using OR.</p>
PSZ	pszDocuments	The name of one or more documents.

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
CONTINUE_RC	The folder export has not completed yet. Call <i>EwfExportFolderFP</i> again.
other	...

Code sample

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

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

// Export the folder SAMPLE1 to path C:\PROJECT with the
// folder Translation Memory and all folder dictionaries,
// overwrite any previously exported folder in path C:\PROJECT
if ( !usRC )
{
do
{
usRC = EqfExportFolderFP( hSession, "SAMPLE1",
'C:\PROJECT',
WITHMEM_OPT | WITHDICT_OPT | OVERWRITE_OPT,
NULL, NULL );
} while ( usRC == CONTINUE_RC );
```

```

} /* endif */

// terminate the session
EqfEndSession( hSession );
}

```

EwfExportFolderFPAs

Purpose

EwfExportFolderFPAs programming interface calls **EwfExportFolderFPAs**. **EwfExportFolderFPAs** exports a folder to a specific path with the option to specify a new filename to the exported folder.

This function performs the export in small units. Call it repetitively until it returns a return code other than CONTINUE_RC.

Format

```

usRC = EwfExportFolderFPAs(hSession,pszFolderName,pszTargetPath,pszExportAs,
WITHMEM_OPT WITHREADONLYMEM_OPT WITHDOCMEM_OPT DELETE_OPT
WITHDICT_OPT OVERWRITE_OPT , ,pszDocuments ,pszDescription,pszMemoryExportAs);

```

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EwfStartSession</i> .
PSZ	pszFolderName	The name of the folder to be exported.
PSZ	pszTargetPath	The path to which the folder is exported.
PSZ	pszExportAs	The filename of the exported folder, or NULL.
LONG	IOptions	<p>The options to be used for the export:</p> <ul style="list-style-type: none"> • WITHMEM_OPT • WITHREADONLYMEM_OPT • WITHDOCMEM_OPT • DELETE_OPT • WITHDICT_OPT • OVERWRITE_OPT <p>These options correspond to those in the "Export Folder" window (see Exporting a folder).</p> <p>You can combine the constants using OR.</p>
PSZ	pszDocuments	The name of one or more documents.
PSZ	pszDescription	The folder description, or NULL.
PSZ	pszMemoryExportAs	The filename of the exported memory in the folder, or NULL.

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
CONTINUE_RC	The folder export has not completed yet. Call <i>EwfExportFolderFP</i> again.
other	...

Code sample

```

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

```

```

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

// Export the folder SAMPLE1 to path C:\PROJECT with the
// folder Translation Memory and all folder dictionaries,
// overwrite any previously exported folder in path C:\PROJECT
// the folder memory is renamed to "MEM1"
if ( !usRC )
{
do
{
usRC = EqfExportFolderFPas( hSession, "SAMPLE1",
'C:\PROJECT', "MyFoll",
WITHMEM_OPT | WITHDICT_OPT | OVERWRITE_OPT,
NULL, NULL, "MEM1" );
} while ( usRC == CONTINUE_RC );
} /* endif */

// terminate the session
EqfEndSession( hSession );
}

```

EwfExportMem

Purpose

EwfExportMem programming interface calls **EwfExportMem**. **EwfExportMem** exports a **Translation Memory** in external format. This function performs the export in small units. Call it repetitively until it returns a return code other than **CONTINUE_RC**.

Format

```
usRC = EwfExportMem(hSession,pszMemName,pszOutFile ,lOptions);
```

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EwfStartSession</i> .
PSZ	pszMemName	The name of the Translation Memory to be exported.
PSZ	pszInFile	The fully qualified name of the file receiving the exported Translation Memory .
LONG	lOptions	The option to be used for the Translation Memory export: <ul style="list-style-type: none"> • OVERWRITE_OPT to replace an existing Translation Memory. • ANSI_OPT (Export in Ansi) • ASCII_OPT (Export in ASCII) • UTF16_OPT (Export in Unicode UTF-16) • TMX_UTF16_OPT (Export in TMX format, use UTF-16 encoding) • TMX_UTF8_OPT (Export in TMX format, use UTF-8 encoding)

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
559 (ERROR_MEM_DATACORRUPT)	The export completed successfully but some characters have been corrupted (i.e. these characters cannot be re-converted to Unicode without loss of data)
CONTINUE_RC	The Translation Memory export has not completed yet. Call <i>EwfExportMem</i> again.

Value	Description
other	...

Code sample

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

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

// Export the Translation Memory MEMDB1 to the external file MEM1.EXP
if ( !usRC )
{
do
{
usRC = EqfExportMem( hSession, "MEMDB1", "C:\\\\MEM1.EXP", 0L );
} while ( usRC == CONTINUE_RC );
} /* endif */

// terminate the session
EqfEndSession( hSession );
}
```

EqfFreeSegFile

Purpose

EqfFreeSegFile programming interface calls **EqfFreeSegFile** Releases the memory occupied by a file loaded into memory using **EqfLoadSegFile**.

Format

usRC = EqfFreeSegFile(hSegFile);

Parameters

Type	Parameter	Description
HPARSESEGFILE	hSegFile	Handle of loaded segmented file

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	Error code (EQF message number). Use EqfGetLastError to retrieve the complete error information.

Code sample

```
{
USHORT usRC = 0;
HPARSSEGFILE *hSegFile = NULL;
HSESSION hSession = 0L;
PARSSEGMENTW Segment;

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

if ( !usRC )
{
usRC = EqfBuildSegDocName( hSession, "SAMPLE1", "Document1",
1, szFileName );
} //endif

if ( !usRC )
{
usRC = EqfLoadSegFile( hSession, szFileName, &hSegFile );
if ( !usRC )
{
usRC = EqfGetSegW( hSegFile, 1, &Segment );
if ( !usRC )
```

```

{
wcsiwr( Segment.szData );
usRC = EqfUpdateSegW( hSegFile, 1, &Segment );
if ( !usRC )
{
usRC = EqfWriteSegFile( hSegFile, szFileName );
} //endif
} //endif
EqfFreeSegFile(hSegFile );
} //endif
} //endif

// terminate the session
EqfEndSession( hSession );
}

```

ErfGetFolderProp

Purpose

ErfGetFolderProp programming interface calls **ErfGetFolderProp**. *ErfGetFolderProp* retrieves the following properties of the specified folder or subfolder:

- Target drive
- Target language
- Name of the read-write memory
- List of read-only memories
- List of dictionaries.

Format

usRC = ErfGetFolderProp(hSession, pszFolderName,); structExtFolProp chDrive,
szTargetLang[MAX_LANG_LENGTH], szRWMemory[MAX_LONGFILESPEC],
szROMemTbl[MAX_NUM_OF_READONLY_MDB][MAX_LONGFILESPEC],
szDicTbl[NUM_OF_FOLDER_DICS][MAX_FILESPEC],

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>ErfStartSession</i> .
PSZ	pszFolderName	The name of the folder.
PEXTFOLPROP	pstructExtFolProp	See ...for details.

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	Error code (EQF message number). Use <i>ErfGetLastError</i> to retrieve the complete error information.

Code sample

```

{
USHORT usRC = 0;
HSESSION hSession = 0L;
EXTFOLPROP FolderProps;

// start the Erf calling interface session
usRC = ErfStartSession(&hSession);

// Retrieve properties of Folder Test.
if ( !usRC )
{
usRC = ErfGetFolderProp(hSession,"Test",&FolderProps);
} //endif

// terminate the session
ErfEndSession( hSession );
}

```

Parameters for structExtFolProp

```
typedef struct _EXTFOLPROP
{
    CHAR chDrive;
    CHAR szTargetLang[MAX_LANG_LENGTH];
    CHAR szRWMemory[MAX_LONGFILESPEC];
    CHAR szROMemTbl[MAX_NUM_OF_READONLY_MDB] [MAX_LONGFILESPEC];
    CHAR szDicTbl[ NUM_OF_FOLDER_DICS] [MAX_FILESPEC];
} _EXTFOLPROP, *PEXTFOLPROP;
```

Type	Parameter	Description
CHAR	chDrive	Returns the target drive of the folder.
CHAR	szTargetLang [MAX_LANG_LENGTH]	Returns the target language.
CHAR	szRWMemory [MAX_LONGFILESPEC]	Returns the read-write memory.
CHAR	szROMemTbl [MAX_NUM_OF_READONLY_MDB] [MAX_LONGFILESPEC]	Returns a list of read-only memories.
CHAR	szDicTbl [NUM_OF_FOLDER_DICS] [MAX_FILESPEC]	Returns the list of dictionaries.

EqfGetLastError

Purpose

EqfGetLastError programming interface calls **EqfGetLastError**. *EqfGetLastError* receives the text of the last error message.

Format

usRC = EqfGetLastError (hSession,pusRc,pszMsgBuf, usBufSize);

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EqfStartSession</i> .
PUSHORT	pusRc	The OpenTM2 return code (message number).
PSZ	pszMsgBuf	An allocated area to receive the message text.
USHORT	usBufSize	The size of the preallocated buffer.

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	Error code (EQF message number).

EqfGetMatchLevel

Purpose

EqfGetMatchLevel programming interface calls **EqfGetMatchLevel**. The API call *EqfGetMatchLevel* computes the match level of the given proposal for the supplied segment. The segment data and the proposal is passed to the function using a EQFSEGINFO structure.

Format

usRC = EqfGetMatchLevel (hSession,pSegment,pProposal, psMatchLevel,pMatchState,lOptions);

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as

Type	Parameter	Description
PEQFSEGINFO	pSegment	<p>returned by <i>EqrStartSession</i> .</p> <p>Pointer to an EQFSEGINFO structure containing the segment data.</p> <p>Note: The target part has not to be filled.</p>
PEQFSEGINFO	pProposal	Pointer to an EQFSEGINFO structure containing the proposal data.
PSHORT	psMatchLevel	Pointer to a SHORT variable receiving the match level. The returned match level is in the range from 0 to 100.
PSHORT	pMatchState	<p>Pointer to a SHORT variable receiving the match state. The returned match state can be:</p> <ul style="list-style-type: none"> • REPLACE_MATCHSTATE for a replace match • FUZZYREPLACE_MATCHSTATE for a fuzzy replace match • FUZZY_MATCHSTATE for a fuzzy match • NONE_MATCH if the proposal is no match at all • EXACT_MATCHSTATE for an exact match • EXACTEXACT_MATCHSTATE for an exact match coming from the same document and same segment.
LONG	IOptions	<p>The options to be used for the function:</p> <ul style="list-style-type: none"> • NO_GENERIC_INLINETAG_REPL_OPT if set the function “generic inline tag replacement” is not used • USE_GENERIC_INLINETAG_REPL_OPT if set the function “generic inline tag replacement” is always used <p>If none of these values is specified, the settings from the “System preferences” are used.</p>

Return code
USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	Error code (EQF message number). Use EQFGetError to retrieve the complete error information.

EQFSEGINFO structure

Type	Field	Description
WCHAR [2048]	szSource	The proposal source in UTF-16 encoding
WCHAR [2048]	szTarget	The proposal target in UTF-16 encoding
LONG	lSegNumber	The segment number
CHAR [256]	szDocument	The name of the document

Type	Field	Description
CHAR [20]	szSourceLanguage	The source language of the proposal
CHAR [20]	szTargetLanguage	The target language of the proposal
CHAR [13]	szMarkup	The name of the markup table

Code sample

```

// the segment data from the document
EQFSEGINFO SegmentData =
{
    L"The <strong>IBM Websphere Translation Server</strong> performs automatic translations.",
    L"",
    1,
    "document.idd",
    "English(U.S.)",
    "German(DPAnat)",
    "IBMIDDOC"
};

// data for a fuzzy match
EQFSEGINFO FuzzyMatch =
{
    L"The <strong>IBM Websphere Translation Server</strong> does automatic translations.",
    L"Der <strong>IBM Websphere Translation Server</strong> macht automatische
Uebersetzungen.",
    7,
    "anotherdoc.idd",
    "English(U.S.)",
    "German(DPAnat)",
    "IBMIDDOC"
};

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

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

    // check the match level of the match in FuzzyMatch
    if ( !usRC )
    {
        SHORT sMatchLevel = 0;
        EQFMATCHSTATE MatchState;

        usRC = EqfGetMatchLevel( hSession, &SegmentData , &FuzzyMatch, &sMatchLevel,
        &MatchState, 0 );
    } /* endif */

    // terminate the session
    EqfEndSession( hSession );
}

```

EqfGetProgress

Purpose

EqfGetProgress programming interface calls **EqfGetProgress** Get the progress of the currently performed function. The progress values returned are in the range from 0 to 100. This API call can only be used for nonDDE API processes which are called repeatedly until the function has been completed (e.g. *EqfImportFolder*).

Format

usRC = EqfGetProgress (hSession,pusProgress);

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EqfStartSession</i> .
PUSHORT	pusProgress	Address of a variable

Type	Parameter	Description
		receiving the current progress value

Return code
USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	Error code (EQF message number). Use <i>EqfGetLastError</i> to retrieve the complete error information.

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

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

if ( !usRC )
{
do
{
usRC = EqfImportFolder( hSession, "SAMPLE1", 'C', '0', WITHMEM_OPT );
if ( usRC == CONTINUE_RC )
{
EqfGetProgress ( hSession, &usProgress );
} //endif
} while ( usRC == CONTINUE_RC );
} // endif

// terminate the session
EqfEndSession( hSession );
}
```

EqfGetSegNum

Purpose

EqfGetSegNum programming interface calls *EqfGetSegNum* Get the number of segments contained in a segmented file loaded into memory using *EqfLoadSegFile*.

Format

usRC = EqfGetSegNum(hSegFile,pISegNum);

Parameters

Type	Parameter	Description
HPARSESEGFILE	hSegFile	Handle of loaded segment file
PLONG	pISegNum	Pointer to a buffer receiving the number of segments in the loaded file

Return code
USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	Error code (EQF message number). Use <i>EqfGetLastError</i> to retrieve the complete error information.

```
{
USHORT usRC = 0;
HPARSSEGFILE *hSegFile = NULL;
HSESSION hSession = 0L;
LONG INumberOfSegments = 0;

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

```

if ( !usRC )
{
usRC = EqfBuildSegDocName( hSession, "SAMPLE1", "Document1",
1, szFileName );
} //endif

if ( !usRC )
{
usRC = EqfLoadSegFile( hSession, szFileName, &hSegFile );
if ( !usRC )
{
usRC = EqfGetSegNum( hSegFile, &lNumberOfSegments);
EqfFreeSegFile(hSegFile );
} //endif
} //endif

// terminate the session
EqfEndSession( hSession );
}

```

EqfGetSegW

Purpose

EqfGetSegW programming interface calls **EqfGetSegW** Get the data of a specific segment from a segmented file loaded into memory using **EqfLoadSegFile**.

Format

usRC = EqfGetSegW(hSegFile,lSegNum,pSeg);

Parameters

Type	Parameter	Description
HPARSESEGFILE	hSegFile	Handle of loaded segment file
LONG	lSegNum	Number of segment being rereived
PPARSESEGMENTW	pSeg	Pointer to structure receiving the segment data

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	Error code (EQF message number). Use EqfGetLastError to retrieve the complete error information.

```

{
USHORT usRC = 0;
HPARSSEGFILE *hSegFile = NULL;
HSESSION hSession = 0L;
PARSSEGMENTW Segment;

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

if ( !usRC )
{
usRC = EqfBuildSegDocName( hSession, "SAMPLE1", "Document1",
1, szFileName );
} //endif

if ( !usRC )
{
usRC = EqfLoadSegFile( hSession, szFileName, &hSegFile );
if ( !usRC )
{
usRC = EqfGetSegW( hSegFile, 1, &Segment );
if ( !usRC )
{
wcslwr( Segment.szData );
}
}
}

```

```

usRC = EqfUpdateSegW( hSegFile, 1, &Segment );
if ( !usRC )
{
usRC = EqfWriteSegFile( hSegFile, szFileName );
} //endif
} //endif
EqfFreeSegFile(hSegFile );
} //endif
} //endif

// terminate the session
EqfEndSession( hSession );
}

```

EqfGetSegmentNumber

Purpose

EqfGetSegmentNumber programming interface calls **EqfGetSegmentNumber**

EqfGetSegmentNumber computes the number of the segment to which the character at the given line and column position belongs to.

Format

usRC = EqfGetSegmentNumber(hSegFile,ILine,IColumn,pISeg);

Parameters

Type	Parameter	Description
HPARSESEGFILE	hSegFile	The handle of a segmented file as returned by function <i>EqfLoadSegFile</i> .
LONG	ILine	Number of the line for which the segment number is requested
LONG	IColumn	Column position of the segment within the line
PLONG	pISeg	Pointer to a LONG buffer which receives the segment number matching the line and column number

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
10009 (NOMATCHINGSEGMENT_RC)	There is no segment with the given position (either the line number or the column number is out of range)
10008 (INVALIDFILEHANDLE_RC)	The file handle hSegFile is invalid
other	Error code (EQF message number). Use function <i>EqfGetLastError</i> to retrieve complete error information.

EqfGetSourceLine

Purpose

EqfGetSourceLine programming interface calls **EqfGetSourceLine** *EqfGetSourceLine* computes the start line and the end line of the given segment based on the linefeeds contained in the document.

Format

usRC = EqfGetSourceLine(hSegFile,ISeg,pISeg,pISegStartLine,pISegEndLine);

Parameters

Type	Parameter	Description
HPARSESEGFILE	hSegFile	The handle of a segmented file as returned by function <i>EqfLoadSegFile</i> .

Type	Parameter	Description
LONG	ISeg	Number of segment for which the source line information is requested
PULONG	pIStartLine	Pointer to a LONG buffer which receives the starting line number of the segment
PULONG	pIEndLine	Pointer to a LONG buffer which receives the end line number of the segment

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
10006 (SEGMENTISJOINED_RC)	The given segment is joined to a previous segment is not visible in the document
10007 (INVALIDSEGMENT_RC)	The given segment number is invalid or out of range
10008 (INVALIDFILEHANDLE_RC)	The file handle hSegFile is invalid
other	Error code (EQF message number). Use function EqfGetLastError to retrieve complete error information

EeqfGetSysLanguage

Purpose

EeqfGetSysLanguage programming interface calls *EeqfGetSysLanguage*. *EeqfGetSys Language* allows to retrieve the currently active default target language of OpenTM2.

Format

usRC = EeqfGetSysLanguage(hSession,pszSysLanguage)

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EeqfStartSession</i> .
PSZ	pszSystemLanguage	Buffer provided to contain the system language string at output. The length of the buffer has to be at least 20 characters.

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	Error code (EQF message number). Use <i>EeqfGetSysLanguage</i> to retrieve the complete error information.

Code sample

```

{
USHORT usRC = 0;
HSESSION hSession = 0L;
CHAR chSystemLanguage[20];

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

// get the system language
if ( !usRC )
{
    usRC = EeqfGetSysLanguage( hSession, chSystemLanguage );
}

```

```

} /* endif */

// terminate the session
EqfEndSession( hSession );
}

```

EqfGetVersion

Purpose

EqfGetVersion programming interface calls **EqfGetVersion**. *EqfGetVersion* retrieves the version info of OpenTM2.

Format

ulVersion = EqfGetVersion()

Parameters

—none —

Return code

ULONG

Value	Description
ulVersion	The version of OpenTM2 in a byte array, see the code sample for details how to access the version info.

Code sample

```

#include <stdlib.h>
#include "eqf_api.h"

int main( int argc, char *argv[], char *envp[] )
{
BYTE abVersion[4];
ULONG ulVersion = EQFGETVERSION();

memcpy( abVersion, &ulVersion, sizeof(ULONG) );

printf( "TM Version %d\n", (short)abVersion[0] );
printf( "TM Release %d\n", (short)abVersion[1] );
printf( "TM Subrelease %d\n", (short)abVersion[2] );
printf( "TM Build %d\n", (short)abVersion[3] );
} /* end of main */

```

EqfImportDoc

Purpose

EqfImportDoc programming interface calls **EqfImportDoc**. *EqfImportDoc* imports one or more documents and sets the document properties to the specified values. The specified values apply to all documents to be imported. If a document needs different settings, for example a different markup, import it separately.

This function performs the import in small units. Call it repetitively until it returns a return code other than CONTINUE_RC.

Format

usRC = EqfImportDoc(hSession,pszFolderName,,pszFiles ,pszTransMem ,pszMarkup ,pszEditor ,pszSourceLanguage ,pszTargetLanguage ,pszAlias ,pszStartPath,pszConversion,,OVERWRITE_OPT);

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EqfStartSession</i> .
PSZ	pszFolderName	The name of the folder receiving the documents.
PSZ	pszStartPath	The start path if the documents are to be imported with relative path information. If a start path is specified, the files in the list only contain the relative path.

Type	Parameter	Description
PSZ	pszFiles	The fully qualified name of the documents to be imported.
PSZ	pszTransMem	The name of the Translation Memory to be used for the document, if different from that of the folder.
PSZ	pszMarkup	The name of the markup table to be used for the document, if different from that of the folder.
PSZ	pszEditor	The name of the editor to be used for the document, if different from that of the folder.
PSZ	pszSourceLanguage	The name of the source language to be used for the document, if different from that of the folder.
PSZ	pszTargetLanguage	The name of the target language to be used for the document, if different from that of the folder.
PSZ	pszAlias	The alias name for the document.
LONG	IOptions	The option to be used for the document import: OVERWRITE_OPT to replace existing documents.
PSZ	pszConversion	Conversion to be used for document or NULL

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
CONTINUE_RC	The document import has not completed yet. Call <i>EqfImportDoc</i> again.
other	...

Code sample

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

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

// Import the documents DOC1.TXT and DOC2.TXT into folder SAMPLE1
// and overwrite any existing documents, the format of the documents
// is to EQFASCII for all other settings the folder settings will be
// used
if ( !usRC )
{
do
{
usRC = EqfImportDoc( hSession, "SAMPLE1", NULL,
"C:\\DOC1.TXT,C:\\DOC2.TXT",
NULL, "EQFASCII", NULL, NULL, NULL, NULL,
OVERWRITE_OPT );
} while ( usRC == CONTINUE_RC );
} /* endif */

// terminate the session
EqfEndSession( hSession );
}
```

EqfImportDict

Purpose

EqfImportDict programming interface calls **EqfImportDict**. *EqfImportDict* imports a dictionary in SGML dictionary.

This function performs the import in small units. Call it repetitively until it returns a return code other than CONTINUE_RC.

Format

```
usRC = EqfImportDict(hSession,pszInFile,pszDictName ,psPassword , IGNORE_OPT  
REPLACE_OPT COMBINE_OPT ASCII_OPT ANSI_OPT UTF16_OPT );
```

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EqfStartSession</i> .
PSZ	pszInFile	The fully qualified name of the SGML file to be imported.
PSZ	pszDictName	The name of the dictionary to be exported.
PSZ	pszPassword	The dictionary password. Only required if the dictionary exists already and is protected.
LONG	IOptions	The options to be used for the merge of entries during the import: <ul style="list-style-type: none"> • IGNORE_OPT • REPLACE_OPT • COMBINE_OPT • ASCII_OPT • ANSI_OPT • UTF16_OPT

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
CONTINUE_RC	The dictionary import has not completed yet. Call <i>EqfImportDict</i> again.
other	...

Code sample

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

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

// Import the SGML file C:\DICT1.SGM into dictionary ENGLGERM
// and replace existing entries with the imported data
if ( !usRC )
{
do
{
usRC = EqfImportDict( hSession, "C:\\DICT1.SGN",
"ENGLGERM", NULL, REPLACE_OPT )
} while ( usRC == CONTINUE_RC );
} /* endif */

// terminate the session
EqfEndSession( hSession );
}
```

EqfImportFolder

Purpose

EqfImportFolder programming interface calls **EqfImportFolder**. *EqfImportFolder* imports a folder from a specific drive to the specified **OpenTM2** drive. The path from which the folder is imported is always \eqf\export.

This function performs the import in small units. Call it repetitively until it returns a return code other than **CONTINUE_RC**.

Format

```
usRC = EqfImportFolder(hSession,pszFolderName,chFromDrive, chToDrive , WITHMEM_OPT  
WITHDICT_OPT XLIFF_OPT );
```

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EqfStartSession</i> .
PSZ	pszFolderName	The name of the folder to be imported.
CHAR	chFromDrive	The drive from which the folder is exported.
CHAR	chToDrive	The target drive for the imported folder. If omitted, the primary EQF drive is used. The drive must be the primary EQF drive or one of the secondary EQF drives defined in the "Configure Drives" window.
LONG	IOptions	<p>The options to be used for the export:</p> <ul style="list-style-type: none"> • WITHMEM_OPT • WITHDICT_OPT • XLIFF_OPT <p>These options correspond to those in the "Import Folder" window (see Importing a folder). You can combine the constants using OR.</p>

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
CONTINUE_RC	The folder import has not completed yet. Call <i>EqfImportFolder</i> again.
other	...

Code sample

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

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

// Import the folder SAMPLE1 from drive C: to the primary Eqf
// system drive, import the folder with Translation Memory databases
// and dictionaries
if ( !usRC )
{
do
{
usRC = EqfImportFolder( hSession, "SAMPLE1", 'C',
'\0', // use primary Eqf drive
WITHDICT_OPT | WITHMEM_OPT );
} while ( usRC == CONTINUE_RC );
} /* endif */
```

```

// terminate the session
EqfEndSession( hSession );
}

```

EqfImportFolderFP

Purpose

EqfImportFolderFP programming interface calls **EqfImportFolderFP**. *EqfImportFolderFP* imports a folder from a specific path to the specified **OpenTM2** drive.

This function performs the import in small units. Call it repetitively until it returns a return code other than **CONTINUE_RC**.

Format

```
usRC = EqfImportFolder(hSession,pszFolderName,pszFromPath, chToDrive , WITHMEM_OPT  
WITHDICT_OPT XLIFF_OPT );
```

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EqfStartSession</i> .
PSZ	pszFolderName	The name of the folder to be imported.
PSZ	pszFromPath	The path from which the folder is exported.
CHAR	chToDrive	The target drive for the imported folder. If omitted, the primary EQF drive is used. The drive must be the primary EQF drive or one of the secondary EQF drives defined in the "Configure Drives" window.
LONG	lOptions	<p>The options to be used for the export:</p> <ul style="list-style-type: none"> • WITHMEM_OPT • WITHDICT_OPT • XLIFF_OPT <p>These options correspond to those in the "Import Folder" window (see Importing a folder).</p> <p>You can combine the constants using OR.</p>

Return code **USHORT**

Value	Description
0 (NO_ERROR)	The function completed successfully.
CONTINUE_RC	The folder import has not completed yet. Call <i>EqfImportFolderFP</i> again.
other	...

Code sample

```

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

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

// Import the folder SAMPLE1 from path C:\PROJECT to the primary
// Eqf system drive, import the folder with Translation Memory
// databases and dictionaries
if ( !usRC )

```

```

{
do
{
usRC = EqfImportFolderFP( hSession, "SAMPLE1", 'C:\PROJECT',
'\0', // use primary Eqf drive
WITHDICT_OPT | WITHMEM_OPT );
} while ( usRC == CONTINUE_RC );
} /* endif */

// terminate the session
EqfEndSession( hSession );
}

```

EqfImportMem

Purpose

EqfImportMem programming interface calls *EqfImportMem*. *EqfImportMem* imports a **Translation Memory** into **OpenTM2**. This function performs the import in small units. Call it repetitively until it returns a return code other than CONTINUE_RC.

Format

```
usRC = EqfImportMem(hSession,pszMemName,pszInFile , ANSI_OPT ASCII_OPT UTF16_OPT
TMX_OPT CLEANRTF_OPT );
```

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EqfStartSession</i> .
PSZ	pszMemName	The name of the Translation Memory to be imported. If a Translation Memory with this name already exists, the imported data is merged into the existing Translation Memory .
PSZ	pszInFile	The fully qualified name of the file containing the Translation Memory data.
LONG	IOptions	The options to be used for the Translation Memory import: <ul style="list-style-type: none"> • ANSI_OPT (Export/Import in Ansi) • ASCII_OPT (Export/Import in ASCII) • UTF16_OPT (Export/Import in Unicode UTF-16) • TMX_OPT (Import in TMX format) • CLEANRTF_OPT can be used together with the TMX_OPT to remove RTF tags from the imported data

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
CONTINUE_RC	The Translation Memory import has not completed yet. Call <i>EqfImportMem</i> again.
other	...

Code sample

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

```

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

// Import the external Translation Memory MEM1.EXP into Translation
// Memory MEMDB1
if ( !usRC )
{
do
{
usRC = EqfImportMem( hSession, "MEMDB1", "C:\\\\MEM1.EXP", 0L );
} while ( usRC == CONTINUE_RC );
} /* endif */

// terminate the session
EqfEndSession( hSession );
}

```

EqfLoadSegFile

Purpose

EqfLoadSegFile programming interface calls **EqfLoadSegFile** Loads a segmented OpenTM2 document file into memory. The segments of the loaded file can be accessed using the **EqfGetSegW** API.

Format

usRC = EqfLoadSegFile(hSegFile,pszFileName);

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by EqfStartSession .
PSZ	pszMemName	Fully qualified file name
HPARSESEGFILE *	hSegFile	Points to the buffer receiving the handle of the loaded segmented file

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	Error code (EQF message number). Use EqfGetLastError to retrieve the complete error information.

Code sample

```

{
USHORT usRC = 0;
HPARSESEGFILE *hSegFile = NULL;
HSESSION hSession = 0L;
PARSSEGMENTW Segment;

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

if ( !usRC )
{
usRC = EqfBuildSegDocName( hSession, "SAMPLE1", "Document1",
1, szFileName );
} //endif

if ( !usRC )
{
usRC = EqfLoadSegFile( hSession, szFileName, &hSegFile );
if ( !usRC )
{
usRC = EqfGetSegW( hSegFile, 1, &Segment );
if ( !usRC )
{
wcslwr( Segment.szData );
usRC = EqfUpdateSegW( hSegFile, 1, &Segment );
}
}
}

```

```

if ( !usRC )
{
    usRC = EqfWriteSegFile( hSegFile, szFileName );
} //endif
} //endif
EqfFreeSegFile(hSegFile );
} //endif
} //endif

// terminate the session
EqfEndSession( hSession );
}

```

EqfOrganizeMem

Purpose

EqfOrganizeMem programming interface calls **EqfOrganizeMem**. *EqfOrganizeMem* organizes the specified **Translation Memory**. This function performs the organization in small units. Call it repetitively until it returns a return code other than **CONTINUE_RC**.

Format

usRC = EqfOrganizeMem(hSession,pszMemName);

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EqfStartSession</i> .
PSZ	pszMemName	The name of the Translation Memory to be organized.

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
CONTINUE_RC	The organization of the Translation Memory has not completed yet. Call <i>EqfOrganizeMem</i> again.
other	http://www.doodle.com/mdzib2ffxmm7zgctbg3a3hbh/admin

Code sample

```

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

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

// organize the Translation Memory MEMDB1
if ( !usRC )
{
do
{
    usRC = EqfOrganizeMem( hSession, "MEMDB1" );
} while ( usRC == CONTINUE_RC );
} /* endif */

// terminate the session
EqfEndSession( hSession );
}

```

EqfProcessNomatch

Purpose

EqfProcessNomatch programming interface calls **EqfProcessNomatch**. The API call *EqfProcessNomatch* reads one or more SNOMATCH files (created using the analysis option "Create file containing untranslated segments") and looks up the segments contained in the SNOMATCH files in the input memory. Each matching proposal (exact and fuzzy match) is written to the output memory. The API call creates a memory match word count and a duplicate word count for the segments in the SNOMATCH files. The word count reports can be created in text and

XML form.

This function performs the processing in small units unless told to complete in one call using the COMPLETE_IN_ONE_CALL_OPT flag. Call it repetitively until it returns a return code other than CONTINUE_RC.

Format

```
usRC = EqfProcessNomatch(hSession,pszNomatch,
pszInMemory,pszOutMemory,pszMemMatchReportText,pszMemMatchReportXml,
pszDupReportText,pszDupReportXml,IOptions);
```

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EwfStartSession</i> .
PSZ	pszNomatch	<p>The specification for the SNOMATCH files to use for the processing. This parameter is evaluated in the following way:</p> <ul style="list-style-type: none"> • the specified value contains wildcard characters the specified value is used as fully qualified search pattern for the SNOMATCH FILES to be used e.g. "C:\EQF\TEST.F00\SNOMAT CHA*.*" • the specified value contains path delimiters the specified value is used as fully qualified name of the SNOMATCH file to process, if the specified value points to a directory all files in the directory are processed e.g. "C:\EQF\TEST.F00\SNOMAT CHFile1.txt", "C:\EQF\TEST.F00\SNOMAT CH" • the value contains no path delimiters the specified value is used as name of a TM folder, all SNOMATCH files contained in the SNOMATCH directory of this folder are processed e.g. "TEST"
PSZ	pszInMemory	The name of the input memory (TM internal)
PSZ	pszOutMemory	The name of an existing or new internal memory receiving the relevant proposals from the input memory
PSZ	pszMemMatchReportText	The fully qualified name for the memory match word count report in text format, specify NULL if no report of this type should be created
PSZ	pszMemMatchReportXml	The fully qualified name for the memory match word count report in XML format, specify NULL if no report of this type should be created
PSZ	pszDupReportText	The fully qualified name for the duplicate word count report in text format, specify NULL if no report of this type

Type	Parameter	Description
		should be created
PSZ	pszDupReportXml	The fully qualified name for the duplicate word count report in XML format, specify NULL if no report of this type should be created
LONG	IOptions	<p>The option(s) to be used for the processing:</p> <ul style="list-style-type: none"> • COMPLETE_IN_ONE_CALL_OPT If set the API call does not return after each processing step but stays in the API call until the function has been completed • RESPECTCRLF_OPT If set memory proposals having different linebreaks are not used as exact match The options can be combined by using the logical OR operator

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
CONTINUE_RC	The SNOMATCH processing is not complete yet. Call <i>EqfProcessNomatch</i> again.
other	■

Code sample

```
HSESSION hSession;
USHORT usRC;

usRC = EqfStartSession( &hSession );
usRC = EqfProcessNomatch( hSession, "TestFolder",
    "PrevMemory", "NewMemory", NULL,
    "C:\Reports\MemMatch.XML", NULL,
    "C:\Reports\Duplicates.XML", COMPLETE_IN_ONE_CALL_OPT );
usRC = EqfEndSession( hSession );
```

The API *EqfProcessNomatch* is called to process all SNOMATCH files of folder "**TestFolder**", the segments are looked up in the memory "**PrevMemory**" and any relevant matches found are written to the memory "**NewMemory**", the memory match count in XML format will be stored under "**C:\Reports\MemMatch.XML**" and the XML duplicate word count will be stored under "**C:\Reports\Duplicates.XML**", the text versions of the reports are not being used. The API call will complete in one call.

EqfRename

Purpose

EqfRename programming interface calls *EqfRename*. *EqfRename* renames a folder, a dictionary or a Translation Memory.

Format

usRC = EqfRename(hSession,usMode,pszOldName,pszNewName,IOptions);

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by <i>EqfStartSession</i> .
USHORT	usMode	Describes the type of object being renamed, valid are RENAME_FOLDER, RENAME_MEMORY or RENAME_DICTIONARY

Type	Parameter	Description
PSZ	pszOldName	The name of the existing folder, dictionary or Translation Memory.
PSZ	pszNewName	The new name for the folder, dictionary or Translation Memory.
LONG	IOptions	Additional options for the rename function: • ADJUSTREFERENCES_OPT to adjust all references to the rename object (valid only for the rename of a Translation Memory)

Return code
USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	Error code (EQF message number). Use EqfGetLastError to retrieve the complete error information.

Code sample

```

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

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

// rename the Translation Memory MyMemory to MyNewMemory and adjust all references
if ( !usRC )
{
    usRC = EqfRename( hSession, RENAME_MEMORY, "MyMemory",
    "MyNewMemory", ADJUSTREFERENCES_OPT );
} /* endif */

// terminate the session
EqfEndSession( hSession );
}

```

EqfSetSysLanguage

Purpose

EqfSetSysLanguage programming interface calls EqfSetSysLanguage. EqfSetSysLanguage sets the default target language for the OpenTM2 system environment. All OpenTM2 internal character conversions (Unicode to ASCII/ANSI, ASCII/ANSI to Unicode) and linguistic functions will use the provided default target language if no other language settings are available.

This happens e.g. during Translation Memory import/export in ASCII. It is a good coding practice to retrieve the default target language first (EqfGetSysLanguage), set the requested default target language, do your processing and reset the default target language to the previously stored value. Using the EqfSetSysLanguage has the same effect as modifying the Default Target Language on the System Preference Dialog via the GUI.

Format

usRC = EqfSetSysLanguage(hSession,pszSystemLanguage)

Parameters

Type	Parameter	Description
HSESSION	hSession	The EQF session handle, as returned by EqfStartSession .
PSZ	pszSystemLanguage	Buffer provided to contain the system language string. The length of the buffer has to be at least 20 characters.

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	Error code (EQF message number). Use <i>EqfSetSysLanguage</i> to retrieve the complete error information.

Code sample

```

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

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

// Set the default target language to be Japanese
if ( !usRC )
{
    usRC = EqfSetSysLanguage( hSession, "Japanese" );

} /* endif */

// terminate the session
EqfEndSession( hSession );
}

```

EqfStartSession**Purpose**

EqfStartSession programming interface calls *EqfStartSession*. *EqfStartSession* prepares the internal data areas for other non-DDE batch function calls. Call it before any other batch function. After you are finished, call the *EqfEndSession* function to clean up all resources.

Format

usRC = EqfStartSession(hSession);

Parameters

Type	Parameter	Description
HSESSION	hSession	The variable receiving the EQF session handle.

Return code

USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	Error code (EQF message number). You cannot use <i>EqfGetLastError</i> to retrieve complete error information if a call to <i>EqfStartSession</i> failed.

EqfUpdateSegW**Purpose**

EqfUpdateSegW programming interface calls *EqfUpdateSegW*. Update the segment data of a specific segment in a segmented file loaded into memory using *EqfLoadSegFile*.

Format

usRC = EqfUpdateSegW(hSegFile,lSegNum,pSeg);

Parameters

Type	Parameter	Description
HPARSESEGFILE	hSegFile	Handle of loaded segmented file

Type	Parameter	Description
LONG	ISegNum	Number of segment being updated
PPARSESEGMENTW	pSeg	Pointer to structure containing the updated segment data

Return code
USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	Error code (EQF message number). Use <i>EqfGetLastError</i> to retrieve the complete error information.

Code sample

```

{
USHORT usRC = 0;
HPARSESEGFILE *hSegFile = NULL;
HSESSION hSession = 0L;
PARSSEGMENTW Segment;

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

if ( !usRC )
{
usRC = EqfBuildSegDocName( hSession, "SAMPLE1", "Document1",
1, szFileName );
} /* endif */

if ( !usRC )
{
usRC = EqfLoadSegFile( hSession, szFileName, &hSegFile );
if ( !usRC )
{
usRC = EqfGetSegW( hSegFile, 1, &Segment );
if ( !usRC )
{
wcslwr( Segment.szData );
usRC = EqfUpdateSegW( hSegFile, 1, &Segment );
if ( !usRC )
{
usRC = EqfWriteSegFile( hSegFile, szFileName );
} //endif
} //endif
EqfFreeSegFile(hSegFile );
} //endif
} //endif

// terminate the session
EqfEndSession( hSession );
}

```

EqfWriteSegFile

Purpose

EqfWriteSegFile programming interface calls **EqfWriteSegFile** Writes a segmented file loaded into memory using **EqfLoadSegFile** back to disk.

Format

usRC = EqfWriteSegFile(hSegFile,pszFileName);

Parameters

Type	Parameter	Description
HPARSESEGFILE	hSegFile	Handle of loaded segmented file
PSZ	pszFileName	Fully qualified file name

Return code
USHORT

Value	Description
0 (NO_ERROR)	The function completed successfully.
other	Error code (EQF message number). Use <i>EqfGetLastError</i> to retrieve the complete error information.

Code sample

```
{  
USHORT usRC = 0;  
HPARSSEGFILE *hSegFile = NULL;  
HSESSION hSession = 0L;  
PARSSEGMENTW Segment;  
  
// start the Eqf calling interface session  
usRC = EqfStartSession(&hSession);  
  
if ( !usRC )  
{  
    usRC = EqfBuildSegDocName( hSession, "SAMPLE1", "Document1",  
    1, szFileName );  
} //endif  
  
if ( !usRC )  
{  
    usRC = EqfLoadSegFile( hSession, szFileName, &hSegFile );  
    if ( !usRC )  
    {  
        usRC = EqfGetSegW( hSegFile, 1, &Segment );  
        if ( !usRC )  
        {  
            wcslwr( Segment.szData );  
            usRC = EqfUpdateSegW( hSegFile, 1, &Segment );  
            if ( !usRC )  
            {  
                usRC = EqfWriteSegFile( hSegFile, szFileName );  
            } //endif  
        } //endif  
        EqfFreeSegFile(hSegFile );  
    } //endif  
} //endif  
  
// terminate the session  
EqfEndSession( hSession );  
}
```

Working with external markup tables

This chapter provides the information required to work with external markup tables. It describes the format of external markup tables so that you can modify them or create new ones. The user exit mechanism of markup tables and its entry points are described to allow for customized processing of documents at different stages. Finally, a parser application programming interface provides some of OpenTM2's internal functions to expand the possibilities of user exits.

The contents of external markup tables are described in terms of the SGML syntax. You should be familiar with SGML to modify or create markup tables. For a complete description of SGML refer to ISO 8879, *Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML)*.

Creating new markup tables

You can create your own markup table by exporting an existing markup table in external SGML format, modifying it with any text editor, and importing it back into **OpenTM2** under a different name. Markup tables need to be available in an SGML-based format to be imported into OpenTM2. Notice that an exported markup table contains only the nondefault entries.

To become familiar with the content of markup tables you might want to export a markup table and study it before you create a new markup table. See [Exporting a markup table](#) for details.

When you have exported one of the markup tables provided by **OpenTM2** you might see a second tag in the second line <SEGMENTEXIT> *userexit* </SEGMENTEXIT>. *userexit* is the name of the

dynamic-link library (DLL) containing the user exit code. This tag is only required if a user exit is to be used. For more information, refer to [Creating user exits for markup tables](#).

Layout and content of a markup table

The general layout and content of a markup table are as follows:

- A markup table must begin with a `<TAGTABLE>` tag and end with a `</TAGTABLE>` tag.
- Following the `<TAGTABLE>` tag are *header tags* that are descriptive or of general purpose for the markup table. These header tags do not declare individual markup data. You can use them to give the markup table a name and a description, to specify a character set for conversion, or to specify substitution characters. Header tags in a markup table are optional. See [...for a list of allowed header tags and a detailed description](#).

An example of a header tag in a markup table is

`<DESCRNAME>descriptive name</DESCRNAME>`, which lets you specify a name for the markup table that is different from its file name.

- Next, a list of *markup tag definitions* follows. These definitions are the core of a markup table. Each definition describes a specific formatting tag, for example, a header tag, or a soft line feed. The definition always includes the name of the markup tag, and either its length or the delimiting characters. A markup tag definition can include further information, for example, whether the text associated with a markup tag needs to be translated. See [...for a list of allowed tags to define a markup tag in detail](#).

A single markup tag definition always starts with the start tag `<TAG>` and ends with the corresponding end tag `</TAG>`. An example of a markup tag definition is:

```
<TAG>
<STRING>[soft line feed]</STRING>
<LENGTH>16</LENGTH>
<TYPE>STNEUTRAL</TYPE>
<SEGINFO>SEGNEUTRAL</SEGINFO>
</TAG>which defines the markup of a soft line feed. The keyword [soft line feed] is defined as <STRING> [soft line feed]</STRING> and has a length of 16 characters.
<TYPE>STNEUTRAL</TYPE> specifies that this markup tag has no influence on segmenting, and
<SEGINFO>SEGNEUTRAL</SEGINFO> specifies that this markup tag does not influence the
segmenting status.
```

- Markup tags often have *attributes* that specify additional characteristics. For example, a markup tag for tables and figures in a document might use a `width` attribute to specify the width of the element. You need to define all attributes of a markup language in your markup table as well. The definition of attributes is similar to the definition of markup tags, except that each attribute definition is enclosed between the `<ATTRIBUTE>` and `</ATTRIBUTE>` tags. See [...for a list of allowed tags to define an attribute in detail](#). An example of an attribute definition is:

```
<ATTRIBUTE>
<STRING>WIDTH=%</STRING>
<ENDDELIM>' \r\n'</ENDDELIM>
</ATTRIBUTE>
```

which defines the markup of a `WIDTH` attribute. Here, you will notice that the keyword `WIDTH` is supposed to be delimited by one of four delimiting characters, as opposed to the previous example, where an explicit length is specified.

In summary, a markup table has the following layout:

```
<TAGTABLE>
Header tags, as required
<TAG>
markup tag definition
</TAG>
:
<TAG>
markup tag definition
</TAG>
<ATTRIBUTE>
attribute definition (optional)
</ATTRIBUTE>
:
<ATTRIBUTE>
attribute definition (optional)
</ATTRIBUTE>
</TAGTABLE>
```

Notice that all entries use the SGML syntax. All SGML tags must be enclosed in "<" and ">". There are always a start tag and an end tag.

Your markup table can contain up to 1000 entries.

An SGML markup tag or attribute must be at least specified with STRING and ENDDELIM, or STRING and LENGTH.

After you have edited the markup table, you can import it into **OpenTM2**. If you import it into an existing markup table, this table is overwritten.

Substitution characters in a markup table

Your markup tag and attribute definitions in a markup table might require that you specify variable parts. An example is the definition of the WIDTH attribute in the previous section (<STRING>WIDTH=%</STRING>). Because a document can contain any value for the WIDTH attribute, the percentage sign % is used as a substitution character.

You can use the following two substitution characters in a markup table:

- The percentage character (%) substitutes any number of characters.
- The question mark (?) substitutes a single character.

The substitution characters do not distinguish between numeric and alphabetic characters.

Note that these substitution characters can be redefined in the markup table header.

SGML tags for markup table header

The following table contains the definition of the SGML tags that you can use in a markup table header.

Table 9. SGML tags for markup table header

SGML tag	Definition
DESCRIPTION	Specifies a markup table description, which is shown in the "Markup Table Properties" window and the "Markup Table List" window.
DESCRNAME	Specifies a descriptive name for this markup table. For example, the specification of <DESCRNAME>ASCII</DESCRNAME> in the markup table EQFASCII would give it the name ASCII. If nothing is specified, the file name of the markup table is used.
CHARSET	Specifies the character set to be used for import and export of documents that use this markup table. The documents will be converted using the selected character set without the need to do the conversion in a user exit. Specify one of the following character sets: <ul style="list-style-type: none">• ASCII• ANSI• UTF8• UNICODE
SINGLESUBST	Specifies the substitution character to use for single character substitution. The default character is ?.
MULTSUBST	Specifies the substitution character to use for multiple character substitution. The default character is % .
USEUNICODE	Specifies whether segmented source and target files in subdirectories SSOURCE and STARGET are stored in Unicode UTF-16 format. Specify one of the following: <ul style="list-style-type: none">• YES• NO This is the default.
REFLOW	Specifies whether CRLF are allowed to be changed during translation or not. EQFMRI is an example of a markup where RELOW is specified and set to NO. Specify one of the following:

SGML tag	Definition
	<ul style="list-style-type: none"> • YES This is the default. • NO
SEGMENTEXIT	Contains the name of the user exit, if the markup table uses one.

SGML tags for markup tags and markup attributes

The following table contains the definition of the SGML tags that you can use to define markup tags and markup attributes in a markup table.

Table 10. SGML tags for markup tags and markup attributes

SGML tag	Definition
STRING	Specifies the name of the markup tag or markup attribute. The specification of STRING is required for an entry in the markup table.
ENDDELIM	Specifies one character as end delimiter of the markup tag or markup attribute, if it has any. You can enter more than one end delimiter. OpenTM2 checks for all possible string combinations to determine the end of the tag or attribute. A string as end delimiter is not possible. When a tag or attribute has an end delimiter, the specification of its length is omitted or can be set to 0. If a tag or attribute has no end delimiter, its length must be specified. The specification of ENDDELIM is required for an entry in the markup table, if LENGTH is not defined.
LENGTH	Defines the length of a markup tag or markup attribute. It must be specified only if the length of the tag or attribute cannot be determined by a delimiter specified by ENDDELIM.
COLPOSITION	Specifies the column position where the markup tag starts. If a markup tag has no special start position and can occur anywhere in a line, COLPOSITION is omitted or can be set to 0. The default is 0.
TYPE	Defines the type of the markup tag. If TYPE is not specified, STDEL is taken as the default. The following types are possible: <ul style="list-style-type: none"> • STDEL Indicates the start of a new text segment. • ENDDEL Indicates the end of a text segment. • SELFC The markup tag is self-contained, that is, it is a text segment by itself. • STNEUTRAL The markup tag is a start tag, which has no influence on segmenting. • ENDNEUTRAL The markup tag is an end tag, which has no influence on segmenting.
SEGINFO	Determines whether the text following the markup tag is to be segmented. If SEGINFO is not specified, SEGNEUTRAL is taken as the default. <ul style="list-style-type: none"> • SEGOFF Sets segmenting off, that is, no segmentation is done until the next markup tag is found that sets segmenting on again. If two tags follow each other that set segmenting off, it needs two tags that set segmenting on to start segmentation again. • SEGON

SGML tag	Definition
	<p>Sets segmenting on again.</p> <ul style="list-style-type: none"> • SEGNEUTRAL Does not influence the segmenting status. • SEGRESET Resets the segmenting status to on, even if the segmenting level requires more than one SEGON tag to set segmentation on. • PROTECTON All following text, including segmentation control flags, is protected until a markup tag with PROTECTOFF is encountered. • PROTECTOFF Turns off text protection. The following text is handled using normal segmentation rules.
ASSTEXT	<p>Defines types of text following the markup tag. If ASSTEXT is not specified, NOEXPL is taken as the default.</p> <ul style="list-style-type: none"> • TSNL Text follows on the same or the next line and will be associated with the markup tag. • TSL Text follows on the same line and will be associated with the markup tag. • NOEXPL No special processing for associated text is required.
ADDINFO	<p>Specifies whether specific text is to be ignored when segments are aligned during the creation of an Initial Translation Memory :</p> <ul style="list-style-type: none"> • 4 Marks the start of an area to be ignored. • 6 Marks the start of an area to be partly ignored. This applies to tags containing a % sign, for example HEADER] %. • 8 Marks the end of an area to be ignored. • 10 Marks the end of an area to be partly ignored. This applies to tags containing a % sign, for example HEADER %.
CLASSID <i>creating tables of contents table of contents creating document creating its table of contents</i>	<p>Specifies how the contents of STRING is handled. The only class is .. . This means that the text specified for STRING becomes an entry of the table of contents that you can display during the translation of a document using the dialog.</p>
ATTRINFO	<p>Specifies whether a markup tag has attached attributes (YES/NO). NO is the default. If YES is specified, the ATTRIBUTE SGML tag must be used to specify the attributes.</p>
TRANSLATEINFO	<p>Specifies whether the segment associated with the markup tag or markup attribute must be translated or not (YES/NO). If TRANSLATEINFO is not specified, NO is taken as the default.</p>

Examples of markup data and corresponding markup tags

If a document contains, for example, [soft line feed] as markup data, it is usually meant as a

Copyright IBM 2010

so-called inline tag, which means that it is contained in the segment. It has no influence on the segmentation of the document. The corresponding markup tag definition in a markup table looks as follows:

```
<TAG>
<STRING>[soft line feed]</STRING>
<LENGTH>16</LENGTH>
<TYPE>STNEUTRAL</TYPE>
<SEGINFO>SEGNEUTRAL</SEGINFO>
</TAG>
```

<STRING>... defines the markup string, and <LENGTH>... specifies its length. Because the length is specified, no ENDDELIM tag is required. <TYPE>STNEUTRAL<... defines that this markup string has no influence on segmentation. All other markup table SGML tags will be set to the default and therefore need not be specified.

Assumed that such markup tag causes segmentation, we define this as follows:

```
<TAG>
<STRING>[soft line feed]</STRING>
<LENGTH>16</LENGTH>
<TYPE>STDEL</TYPE>
<SEGINFO>SEGNEUTRAL</SEGINFO>
</TAG>
```

The following table lists some imaginary markup data with a description.

Markup data	Definition
[bold] text [bold]	The text following this tag (until the end tag) is printed bold; this tag is part of the segment and has no influence on segmenting.
[Heading x]text	This tag describes a heading; the heading text must follow on the same line; x is the level of heading and goes from 1 to 9; this tag ends the previous segment and starts a new segment.
[page: even]	A page break; the following text starts on an even page; this tag always starts on the first column and has no text following in the same line; a blank must separate the attribute even from the tag.
[page: odd]	A page break; the following text starts on an odd page; this tag always starts on the first column and has no text following in the same line; a blank must separate the attribute odd from the tag.
[paragraph]	A paragraph; this tag ends the previous segment and starts a new segment; the tag occurs at the end of the previous paragraph.
%	Stands for any number of characters. For example, in b%, % stands for the characters old.
[break]	Starts a new segment. You use this tag to split an existing segment into two or more segments.
[*%]	* indicates the start of a comment and % stands for the comment text.

This markup data would lead to the following markup table definitions. The defaults will not be shown.

Markup definition	Explanation
<pre><TAG> <STRING>[bold]</STRING> <LENGTH>6</LENGTH> <TYPE>STNEUTRAL</TYPE> </TAG> or <TAG> <STRING>[bold</STRING> <ENDDELIM>]</ENDDELIM> <TYPE>STNEUTRAL</TYPE></pre>	The markup tag should be part of the segment, therefore STNEUTRAL is used. All examples have the same result, you can specify this markup tag by its length or end delimiter. You can also substitute part of the inline tag by %.

Markup definition	Explanation
<pre data-bbox="160 196 795 478"></TAG> or <TAG> <STRING>[b%</STRING> <ENDDELIM>]</ENDDELIM> <TYPE>STNEUTRAL</TYPE> </TAG></pre>	
<pre data-bbox="160 478 795 733"><TAG> <STRING>[Heading ?</STRING> <ENDDELIM>]</ENDDELIM> <SEGINFO>SEGRESET</SEGINFO> <ASSTEXT>TSL</ASSTEXT> <TRANSLATEINFO>YES</TRANSLATEINFO> > </TAG></pre>	Single substitution is used for the heading level; the end of the tag is]; the heading requires the reset of segmenting with SEGRESET; the text associated with the tag occurs on the same line; the text associated with the tag is translatable.
<pre data-bbox="160 733 795 935"><TAG> <STRING>[page:</STRING> <ENDDELIM>]</ENDDELIM> <ATTRINFO>YES</ATTRINFO> <COLPOSITION>1</COLPOSITION> </TAG></pre>	The markup tag ends with a blank; attributes may follow; the tag always starts at the first column in a line.
<pre data-bbox="160 935 795 1096"><TAG> <STRING>[paragraph</STRING> <ENDDELIM>]</ENDDELIM> <TYPE>ENDDEL</TYPE> </TAG></pre>	The tag ends with] or is defined by its length; the tag should end the previous segment, therefore ENDDEL is used.
<pre data-bbox="160 1096 795 1352">or <TAG> <STRING>[paragraph]</STRING> <LENGTH>11</LENGTH> <TYPE>ENDDEL</TYPE> </TAG></pre>	
<pre data-bbox="160 1352 795 1486"><ATTRIBUTE> <STRING>even</STRING> <ENDDELIM>]</ENDDELIM> </ATTRIBUTE></pre>	This is an attribute; it ends with].
<pre data-bbox="160 1486 795 1620"><ATTRIBUTE> <STRING>odd</STRING> <ENDDELIM>]</ENDDELIM> </ATTRIBUTE></pre>	This is an attribute; it ends with].
<pre data-bbox="160 1620 795 1782"><TAG> <STRING>[break]</STRING> <LENGTH>7</LENGTH> <TYPE>STDEL</TYPE> </TAG></pre>	Indicates that a new segment starts.
<pre data-bbox="160 1782 795 1943"><TAG> <STRING>*%</STRING> <ENDDELIM>\r\n</ENDDELIM> <COLPOSITION>1</COLPOSITION> </TAG></pre>	Indicates a comment that ends at the end of the line. COLPOSITION defines that the asterisk is only recognized as the start of a comment if it appears in the first column of a line.

Creating user exits for markup tables

There are document formats that require a user exit for their markup table:

- Binary documents, for example Microsoft^(R) Word for Windows^(R) documents
- Documents that require code page conversion, for example ANSI documents
- Documents that have a fixed record layout
- Documents that contain nontranslatable text parts, for example, RTF documents
- Binary documents like Lotus Notes database files and template files that require context-dependent processing.

OpenTM2 provides two markup tables that are already combined with a user exit:

- The user exit part of the EQFHTML4 markup table converts the code page and preprocesses JavaScripts to limit segments to 2048 characters. The markup table part controls text

- segmentation and the recognition of inline tags.
- The user exit part of the EQFANSI markup table converts the code page, and the markup table part inserts segment breaks after empty lines.

In addition, **OpenTM2** provides a user exit that you can use with the appropriate markup table. This user exit is a dynamic-link library (DLL) with predefined entry points. The code for the exit can be written in any programming language that supports PASCAL calling conventions. The include file `EQF_API.H` contains the definitions required for a user exit written in C.

The user exit is activated using the `<SEGMENTEXIT>` tag of the markup table (see also [Segment exit](#)).

General user exit entry points

The user exit entry points (their names start with EQF) are called at different stages during the analysis, translation, and export of a document.

- During the analysis (see `...`):
- `...` is called *before* the text is segmented. It can be used to preprocess a document and decide whether text segmentation is done by OpenTM2 after EQFPRESEG2.

`user exitentry pointEQFPRESEGW`

- `...` is called *after* the text is segmented. It can be used to postprocess a document.

`user exitentry pointEQFPOSTSEGW`

- `...` is called *after* Translation Memory matches are processed and terms lists are created. It can be used to modify segments.

`user exitentry pointEQFPOSTTMW`

Figure 164. Analysis of a document using the user exit

[PIC]Figure shows Analysis of a document using the user exit

- During the translation:
- `...` is called *after* a segment is translated but before it is saved in the Translation Memory. It can be used to modify a segment.

`user exitentry pointEQFCHECKSEGW`

- `...` is called when the user selects the "Show translation" menu item.

`user exitentry pointEQFSHOW`

- During the export (see `...`):
- `...` is called *before* OpenTM2 removes the segmentation from a document. It can be used for the same purpose, or whatever is required at this step.

`user exitentry pointEQFPREUNSEGW`

- `...` is called *after* OpenTM2 (or EQFPREUNSEG2) removed the segmentation. It can be used, for example, to establish the external document format.

`user exitentry pointEQFPOSTUNSEG2`

- Alternatively, `...` can be called *after* OpenTM2 (or EQFPREUNSEG2) removed the segmentation. If EQFPOSTUNSEGW entry point exists, OpenTM2 uses EQFPOSTUNSEGW, without regard of the existence of EQFPOSTUNSEG2. EQFPOSTUNSEGW requires that the input text is always UTF16. If EQFPOSTUNSEGW entry point exists, OpenTM2's "Undo text segmentation" step outputs an UTF16 file.

Figure 165. Export of a document using the user exit

[PIC]Figure shows REQTEXT

The following sections describe the individual entry points in detail. Note that entry points from earlier versions of OpenTM2 (without the trailing letter W) are supported, and the calling syntax remains unchanged. However, you should use the entry points as listed in this section. See [Compatibility notes concerning Unicode support](#) for details.

`programming interface callsEQFPRESEG2` `EQFPRESEG2` is called during the analysis of a document before the text is segmented. It preprocesses the document, for example converts code pages, and decides whether text segmentation is done by **OpenTM2** or `EQFPRESEG2` itself. If an error occurs, it can stop the analysis.

EQFPRESEG2

Purpose

`programming interface callsEQFPRESEG2` `EQFPRESEG2` is called during the analysis of a document before the text is segmented. It preprocesses the document, for example converts code pages, and decides whether text segmentation is done by **OpenTM2** or `EQFPRESEG2` itself. If an error occurs, it can stop the analysis.

Format

`EQFPRESEG2(MarkupTable,Editor,Path,SourceFile,Buffer,OutputFlag,SliderWindowHandle,ReturnFlag)`

Parameters

- *MarkupTable*
The pointer to the name of a markup table.
- *Editor*
The pointer to the name of the editor.
- *Path*
The pointer to the program path.
- *SourceFile*
The pointer to the name of the source file (with full path).
- *Buffer*
The pointer to the buffer containing the name of the temporary output file.
- *OutputFlag*
The output flag indicating whether the text is to be segmented by EQFPRESEG2 instead of **OpenTM2**.
- *SliderWindowHandle*
The handle of the slider window.
- *ReturnFlag*
The pointer to the return flag. If this flag changes to TRUE, the user exit must return immediately.

EQFPRESEGEX

Purpose

programming interface calls EQFPRESEGEX. EQFPRESEGEX is called during the analysis of a document before the text is segmented. It preprocesses the document, for example converts code pages, and decides whether text segmentation is done by OpenTM2 or EQFPRESEGEX itself. If an error occurs, it can stop the analysis. The EQFPRESEGEX entry point is identical to [EQFPRESEG2](#) except for the additional parameter Analysysis handle.

Format

EQFPRESEGEX(*MarkupTable, Editor, Path, SourceFile, Buffer, OutputFlag, SliderWindowHandle, ReturnFlag, AnalysisHandle***)**

Parameters

- *MarkupTable*
The pointer to the name of a markup table.
- *Editor*
The pointer to the name of the editor.
- *Path*
The pointer to the program path.
- *SourceFile*
The pointer to the name of the source file (with full path).
- *Buffer*
The pointer to the buffer containing the name of the temporary output file.
- *OutputFlag*
The output flag indicating whether the text is to be segmented by EQFPRESEGEX instead of **OpenTM2**.
- *SliderWindowHandle*
The handle of the slider window.
- *ReturnFlag*
The pointer to the return flag. If this flag changes to TRUE, the user exit must return immediately.
- *AnalysisHandle*
The analysis handle. This handle is required for the API calls [EQFSETTAOPTIONS](#) and [EQFGETTAOPTIONS](#).

EQFPOSTSEGW

Purpose

programming interface calls EQFPOSTSEGW EQFPOSTSEGW EQFPOSTSEGW is called during the analysis of a document after the text is segmented. It postprocesses the document, for example adjusts segment boundaries. If an error occurs, it can stop the analysis.

Format

EQFPOSTSEGW(*MarkupTable*,*Editor*,*Path*,*SourceFile*,
TargetFile,*SegmentationTags*,*SliderWindowHandle*,*ReturnFlag*)

Parameters

- ***MarkupTable***
The pointer to the name of a markup table.
- ***Editor***
The pointer to the name of the editor.
- ***Path***
The pointer to the program path.
- ***SourceFile***
The pointer to the name of the source file (with full path).
- ***TargetFile***
The pointer to the name of the target file.
- ***SegmentationTags***
The pointer to the tags inserted during text segmentation.
- ***SliderWindowHandle***
The handle of the slider window.
- ***ReturnFlag***
The pointer to the return flag. If this flag changes to TRUE, the user exit must return immediately.

EQFPOSTSEGWEX

Purpose

programming interface calls EQFPOSTSEGWEX EQFPOSTSEGWEX EQFPOSTSEGWEX is called during the analysis of a document after the text is segmented. It postprocesses the document, for example adjusts segment boundaries. If an error occurs, it can stop the analysis. The EQFPOSTSEGWEX entry point is identical to [EQFPOSTSEGW](#) except for the additional parameter Analysis handle.

Format

EQFPOSTSEGWEX(*MarkupTable*,*Editor*,*Path*,*SourceFile*,
TargetFile,*SegmentationTags*,*SliderWindowHandle*,*ReturnFlag*,*AnalysisHandle*)

Parameters

- ***MarkupTable***
The pointer to the name of a markup table.
- ***Editor***
The pointer to the name of the editor.
- ***Path***
The pointer to the program path.
- ***SourceFile***
The pointer to the name of the source file (with full path).
- ***TargetFile***
The pointer to the name of the target file.
- ***SegmentationTags***
The pointer to the tags inserted during text segmentation.
- ***SliderWindowHandle***
The handle of the slider window.

- *ReturnFlag*
The pointer to the return flag. If this flag changes to TRUE, the user exit must return immediately.
- *AnalysisHandle*
The analysis handle. This handle is required for the API calls [EQFSETTAOPTIONS](#) and [EQFGETTAOPTIONS](#).

EQFPOSTTMW

Purpose

programming interface callsEQFPOSTTMW EQFPOSTTMW *EQFPOSTTMW* is called during the analysis of a document after **Translation Memory** matches have been inserted and terms lists have been created. It is used to modify the segments. If an error occurs, it can stop the analysis.

Format

EQFPOSTTMW(*Editor*,*Path*,*SegmentedSourceFile*,
SegmentedTargetFile,*SegmentationTags*,*SourceTargetFlag*,*SliderWindowHandle*,*ReturnFlag*)

Parameters

- *Editor*
The pointer to the name of the editor.
- *Path*
The pointer to the program path.
- *SegmentedSourceFile*
The pointer to the name of the segmented source file.
- *SegmentedTargetFile*
The pointer to the name of the segmented target file.
- *SegmentationTags*
The pointer to the tags inserted during text segmentation.
- *SourceTargetFlag*
The flag indicating if the segmented source differs from the segmented target.
- *SliderWindowHandle*
The handle of the slider window.
- *ReturnFlag*
The pointer to the return flag. If this flag changes to TRUE, the user exit must return immediately.

EQFCHECKSEGW

Purpose

programming interface callsEQFCHECKSEGW EQFCHECKSEGW *EQFCHECKSEGW* is called during the translation of a document after a segment has been translated but not saved yet in the **Translation Memory**. It can modify the segment, for example change lowercase characters to uppercase, and prevent the segment from being saved, for example if specific length limits have been exceeded.

EQFCHECKSEGW is also called when exact matches are automatically substituted during the analysis of a document.

Format

EQFCHECKSEGW(*PreviousSourceSegment*,*CurrentSourceSegment*,
Translation,*ModifyFlag*,*MessageFlag*)

Parameters

- *PreviousSourceSegment*
The pointer to the text of the previous source segment.
- *CurrentSourceSegment*
The pointer to the text of the current source segment.
- *Translation*
The pointer to the translation of the current segment.

- *ModifyFlag*
The pointer to the flag that is set when the user exit has modified the translated segment.
- *MessageFlag*
The flag indicating whether a message box is shown.

Return code

The return code indicates if the segment can be saved.

EQFSHOW

Purpose

programming interface callsEQFSHOW EQFSHOW is called during the translation of a document when the user selects the "Show Translation" menu item. It is up to the user exit to prepare and display the document in a window. The user exit can use the API calls **EQFGETNEXTSEG**, **EQFGETNEXTSEGW**, **EQFGETPREVSEG**, **EQFGETPREVSEGW**, **EQFGETCURSEG**, **EQFGETCURSEGW** and **EQFGETINFO** to retrieve the document segments and to get other document information.

Format

EQFSHOW(lInfo,hwndParent)

Parameters

- *lInfo*
A handle to the target document. This handle has to be specified in the API calls for accessing the segment text.
- *hwndParent*
The handle of the window which should be specified as parent window for the window displaying the document.

Return code

The user exit should return TRUE if the document could be displayed and FALSE in case of errors.

EQFGETCURSEG

Purpose

programming interface callsEQFGETCURSEG EQFGETCURSEG returns a specific segment from the document identified by the lInfo handle. The text of the segment is stored in the buffer pointed to by pBuffer as a zero terminated string. The variable pointed to by pusSegNum contains the number of the requested segment.

Format

EQFGETCURSEG(lInfo,pusSegNum,pBuffer,pusBufSize)

Parameters

- *lInfo*
The document handle which has been passed to the user exit as the first parameter of the EQFSHOW entry point.
- *pusSegNum*
The pointer to a ULONG variable containing the segment number.
- *pBuffer*
The pointer to a buffer for the segment text.
- *pusBufSize*
The pointer to a USHORT variable containing the size of the buffer pointed to by pBuffer.

Return code

The function returns zero if successful otherwise an error code is returned.

EQFGETCURSEGW

Purpose

programming interface callsEQFGETCURSEGW EQFGETCURSEGW returns a specific segment from the document identified by the lInfo handle. The text of the segment is stored in the buffer pointed to by pBuffer in UTF16-encoding and is terminated by 0x0000. The variable pointed to by pulSegNum contains the number of the requested segment.

Format

EQFGETCURSEGW(lInfo,pulSegNum,pBuffer,pusBufSize)

Parameters

- *lInfo*
The document handle which has been passed to the user exit as the first parameter of the EQFSHOW entry point.
- *pulSegNum*
The pointer to a ULONG variable containing the segment number.
- *pBuffer*
The pointer to a buffer for the segment text in UTF-16 encoding.
- *pusBufSize*
The pointer to a USHORT variable containing the size of the buffer pointed to by pBuffer in number of UTF-16 characters.

Return code

The function returns zero if successful otherwise an error code is returned.

EQFGETNEXTSEG

Purpose

programming interface callsEQFGETNEXTSEG EQFGETNEXTSEG EQFGETNEXTSEG returns the next segment from the document identified by the lInfo handle. The text of the segment is stored in the buffer pointed to by pBuffer as a zero-terminated string. The API call increments the segment number automatically.

Format

EQFGETNEXTSEG(lInfo,pulSegNum,pBuffer,pusBufSize)

Parameters

- *lInfo*
The document handle which has been passed to the user exit as the first parameter of the EQFSHOW entry point.
- *pulSegNum*
The pointer to a USHORT variable containing the segment number. This variable should be set to 1 before the first call. The segment number is automatically incremented.
- *pBuffer*
The pointer to a buffer for the segment text.
- *pusBufSize*
The pointer to a USHORT variable containing the size of the buffer pointed to by pBuffer.

Return code

The function returns zero if successful otherwise an error code is returned.

EQFGETNEXTSEGW

Purpose

programming interface callsEQFGETNEXTSEGW EQFGETNEXTSEGW EQFGETNEXTSEGW returns the next segment from the document identified by the lInfo handle. The text of the segment is stored in the buffer pointed to by pBuffer in UTF-16 encoding and is terminated by 0x0000. The API call increments the segment number automatically.

Format

EQFGETNEXTSEGW(lInfo,pulSegNum,pBuffer,pusBufSize)

Parameters

- *lInfo*
The document handle which has been passed to the user exit as the first parameter of the EQFSHOW entry point.
- *pulSegNum*
The pointer to a ULONG variable containing the segment number. This variable should be set to 1 before the first call. The segment number is automatically incremented.
- *pBuffer*
The pointer to a buffer for the segment text in UTF-16 encoding.
- *pusBufSize*

The pointer to a USHORT variable containing the size of the buffer in number of UTF-16 characters.

Return code

The function returns zero if successful otherwise an error code is returned.

EQFGETPREVSEG

Purpose

programming interface callsEQFGETPREVSEG EQFGETPREVSEG EQFGETPREVSEG returns the previous segment from the document identified by the lInfo handle. The text of the segment is stored in the buffer pointed to by pBuffer as a zero-terminated string. The API call decrements the segment number automatically.

Format

EQFGETPREVSEG(lInfo,pusSegNum,pBuffer,pusBufSize)

Parameters

- *lInfo*
The document handle which has been passed to the user exit as the first parameter of the EQFSHOW entry point.
- *pulSegNum*
The pointer to a USHORT variable containing the segment number. The segment number is automatically decremented.
- *pBuffer*
The pointer to a buffer for the segment text.
- *pusBufSize*
The pointer to a USHORT variable containing the size of the buffer pointed to by pBuffer.

Return code

The function returns zero if successful otherwise an error code is returned.

EQFGETPREVSEGW

Purpose

programming interface callsEQFGETPREVSEGW EQFGETPREVSEGW EQFGETPREVSEGW returns the previous segment from the document identified by the lInfo handle. The text of the segment is stored in the buffer pointed to by pBuffer in UTF16-encoding and is terminated by 0x0000. The API call decrements the segment number automatically.

Format

EQFGETPREVSEGW(lInfo,pulSegNum,pBuffer,pusBufSize)

Parameters

- *lInfo*
The document handle which has been passed to the user exit as the first parameter of the EQFSHOW entry point.
- *pulSegNum*
The pointer to a ULONG variable containing the segment number. The segment number is automatically decremented.
- *pBuffer*
The pointer to a USHORT variable containing the size of the buffer pointed to by pBuffer in number of UTF-16 characters.
- *pusBufSize*
The pointer to a USHORT variable containing the size of the buffer pointed to by pBuffer.

Return code

The function returns zero if successful otherwise an error code is returned.

EQFBUILDDOCPATH

Purpose

programming interface callsEQFBUILDDOCPATH EQFBUILDDOCPATH EQFBUILDDOCPATH creates the fully qualified file name for a OpenTM2 document using the folder object name and the document long name. This function can be used to access documents stored in OpenTM2 folders.

Format

EQFBUILDDOCPATH(szFolObjName,szDocLongName,PathID,pchBuffer)

Parameters

- **szFolObjName**
The folder object name as returned using EQFGETINFO with the GETINFO_FOLDEROBJECT ID.
- **szDocLongName**
The document long name.
- **PathID**
The ID of the requested document path, valid IDs are:PATHID_SOURCE to build the path to the source document
PATHID_SEGSOURCE to build the path to the segmented source document
PATHID_SEGTARGET to build the path to the segmented target document
PATHID_TARGET to build the path to the target document
- **pchBuffer**
The pointer to a buffer receiving the fully qualified document path, the size of this buffer has to be at least 60 bytes.

Return code

- **0**
function completed successfully
- **ERROR_INVALID_PARAMETER**
wrong or missing parameter
- **ERROR_PATH_NOT_FOUND**
the folder did not exist
- **ERROR_FILE_NOT_FOUND**
the document does not exist

Examples

The folder "AnotherTestFolder" contains the document "myTest.HTML". The folder is located on drive "E:" and has a short name of "ANOTH000.F00". The document short name is "MYTESTHT.000". The primary drive of the OpenTM2 installation is "C:".

EQFBUILDDOCPATH("C:\EQF\ANOTH000.F00", "myTest.HTML", PATHID_SOURCE, szBuffer) would return " E:\EQF\ANOTH000.F00\SOURCE\ MYTESTHT.000" in szBuffer.

EQFGETINFO

Purpose

programming interface callsEQFGETINFO EQFGETINFO returns specific on the document currently being processed in the EQFSHOW function of the user exit. This function is used by the user exit to get more information concerning the document and its location.

Format

EQFGETINFO(lInfo,InfoID,pchBuffer,pusBufSize)

Parameters

- **lInfo**
The info handle passed to the user exit in the EQFSHOW call.
- **InfoID**
The ID of the requested information, valid IDs are:GETINFO_MARKUP to retrieve the markup table of the document
GETINFO_FOLDEROBJECT to retrieve the object name of the folder containing the document
GETINFO_FOLDERLONGNAME to retrieve the long name (in ASCII) of the folder containing the document
GETINFO_DOCFULLPATH to retrieve the fully qualified path of the document segmented target file
GETINFO_DOCLONGNAME to retrieve the document long name
- **pchBuffer**
The pointer to a buffer receiving the requested information, if this parameter is NULL the size of the requested information is returned using the pusBufSize parameter.

- ***pusBufSize***
The pointer to a USHORT value containing the buffer size, on return this value contains the size of the returned information.

Return code

- **0**
function completed successfully
- **ERROR_INVALID_PARAMETER**
unknown InfoID or missing parameter
- **ERROR_INVALID_HANDLE**
invalid Info handle
- **ERROR_NOT_ENOUGH_MEMORY**
not enough memory / memory allocation failed
- **ERROR_INSUFFICIENT_BUFFER**
buffer is not large enough for the returned information, *pusBufSize contains required buffer size

Examples

Assuming the document "myTest.HTML" located in folder "AnotherTestFolder" is opened using EQFSHOW. The folder is located on drive "E:" and has a short name of "ANOTH000.F00". The document short name is "MYTESTHT.000". The primary drive of the OpenTM2 installation is "C:"

usBufSize = sizeof(szBuffer); EQFGETINFO(lInfo, GETINFO_MARKUP, szBuffer, &usBufSize) would return "IBMHTM32" in szBuffer

usBufSize = sizeof(szBuffer); EQFGETINFO(lInfo, GETINFO_FOLDEROBJECT, szBuffer, &usBufSize) would return "C:\EQF\ANOTH000.F00" in szBuffer

usBufSize = sizeof(szBuffer); EQFGETINFO(lInfo, GETINFO_FOLDERLONGNAME, szBuffer, &usBufSize) would return "AnotherTestFolder" in szBuffer

usBufSize = sizeof(szBuffer); EQFGETINFO(lInfo, GETINFO_DOCFULLPATH, szBuffer, &usBufSize) would return "E:\EQF\ANOTH000.F00\STARGET\MYTESTHT.000" in szBuffer

usBufSize = sizeof(szBuffer); EQFGETINFO(lInfo, GETINFO_DOCLONGNAME, szBuffer, &usBufSize) would return "MyTest.HTML" in szBuffer

EQFPREUNSEGW

Purpose

programming interface calls EQFPREUNSEGW. EQFPREUNSEGW is called during the export of a document before the segmentation tags inserted by OpenTM2 are removed. It decides whether the segmentation tags are removed by OpenTM2 or EQFPREUNSEGW itself. However, it is normally used to remove the segmentation tags. If an error occurs, it can stop the export.

Format

**EQFPREUNSEGW(*Editor*,*Path*,*SegmentedTargetFile*,*Buffer*,
SegmentationTags,*OutputFlag*,*SliderWindowHandle*,*ReturnFlag*)**

Parameters

- ***Editor***
The pointer to the name of the editor.
- ***Path***
The pointer to the program path.
- ***SegmentedTargetFile***
The pointer to the name of the segmented target file (with full path).
- ***Buffer***
The pointer to the buffer containing the name of the temporary output file.
- ***SegmentationTags***
The pointer to the tags inserted during text segmentation.

- *OutputFlag*
The output flag indicating whether the segmentation tags are removed by EQFPREUNSEGW instead of OpenTM2.
- *SliderWindowHandle*
The handle of the slider window.
- *ReturnFlag*
The pointer to the return flag. If this flag changes to TRUE, the user exit must return immediately.

EQFPOSTUNSEGW

Purpose

programming interface callsEQFPOSTUNSEGW EQFPOSTUNSEGW EQFPOSTUNSEGW is called during the export of a document after the segmentation tags have been removed from the text. The text must be in UTF16. It is normally used to establish the external document format. If an error occurs, it can stop the export.

Format

EQFPOSTUNSEGW(*MarkupTable,Editor,Path,TargetFile, SegmentationTags,ReturnFlag*)

Parameters

- *MarkupTable*
The pointer to the name of a markup table.
- *Editor*
The pointer to the name of the editor.
- *Path*
The pointer to the program path (with full path).
- *TargetFile*
The pointer to the name of the target file (with full path).
- *SegmentationTags*
The pointer to the tags inserted during text segmentation.
- *ReturnFlag*
The pointer to the return flag. If this flag changes to TRUE, the user exit must return immediately.

EQFPOSTUNSEG2

Purpose

programming interface callsEQFPOSTUNSEG2 EQFPOSTUNSEG2 EQFPOSTUNSEG2 is called during the export of a document after the segmentation tags have been removed from the text. It is normally used to establish the external document format. If an error occurs, it can stop the export.

Format

EQFPOSTUNSEG2(*MarkupTable,Editor,Path,TargetFile, SegmentationTags,ReturnFlag*)

Parameters

- *MarkupTable*
The pointer to the name of a markup table.
- *Editor*
The pointer to the name of the editor.
- *Path*
The pointer to the program path (with full path).
- *TargetFile*
The pointer to the name of the target file (with full path).
- *SegmentationTags*
The pointer to the tags inserted during text segmentation.
- *ReturnFlag*
The pointer to the return flag. If this flag changes to TRUE, the user exit must return immediately.

API calls for user exits

This group contains the API calls which can be called by the markup table user exits to access and modify OpenTM2 settings. Currently these are

- ...to get the active analysis settings. This API call can be called by the user exit during the ..., and ...processing.
- to modify the analysis settings. This API call can be called by the user exit during the ..., and ...processing.

The following sections describe the individual API calls in detail.

EQFGETTAOPTIONS

Purpose

programming interface calls EQFGETTAOPTIONS. EQFGETTAOPTIONS can be used by the markup table user exit to retrieve the currently active analysis settings. The settings are returned in an [EQFTAOPTIONS](#) structure. The analysis handle used by this call is passed to the user exit by the user exit entry points [EQFPRESEGEX](#), and [EQFPOSTSEGWEX](#).

Format

EQFGETTAOPTIONS(*AnalysisHandle*,*Options*)

Parameters

- *AnalysisHandle*
The analysis handle passed to the user exit by the entry points [EQFPRESEGEX](#), and [EQFPOSTSEGWEX](#).
- *Options*
The pointer to a [EQFTAOPTIONS](#) structure receiving the currently active analysis settings.

EQFSETTAOPTIONS

Purpose

programming interface calls EQFSETTAOPTIONS. EQFSETTAOPTIONS can be used by the markup table user exit to change the currently active analysis settings. The settings are passed to the API call in an [EQFTAOPTIONS](#) structure. The analysis handle used by this call is passed to the user exit by the user exit entry points [EQFPRESEGEX](#), and [EQFPOSTSEGWEX](#).

Format

EQFSETTAOPTIONS(*AnalysisHandle*,*Options*)

Parameters

- *AnalysisHandle*
The analysis handle passed to the user exit by the entry points [EQFPRESEGEX](#), and [EQFPOSTSEGWEX](#).
- *Options*
The pointer to a [EQFTAOPTIONS](#) structure containing the analysis settings being modified.

EQFTAOPTIONS

Purpose

programming interface calls EQFTAOPTIONS. The structure [EQFTAOPTIONS](#) is used by the API calls [EQFSETTAOPTIONS](#) and [EQFGETTAOPTIONS](#) to get or set the analysis options.

Fields

- *fAdjustLeadingWS*
This flag represents the "Adjust leading whitespace to whitespace of source segment" flag of the GUI.
- *fAdjustTrailingWS*
This flag represents the "Adjust trailing whitespace to whitespace of source segment" flag of the GUI.
- *bForFutureUse*
Area for future enhancements. Currently not in use.

User exit entry points for context-dependent translations

Lotus Notes markup table with user exit for LOTUSNGD markup table. The following user exit entry points support context-dependent translations, where translation proposals and automatic translations not only depend on text matches but also on the type of document containing the text. These entry points are designed to support the translation of Lotus Notes and Domino design elements, such as Notes database files, template files, and application templates. When OpenTM2 imports these documents (using the LOTUSNGD markup table), it maintains context-dependent information about these design elements together with existing translations in the Translation Memory. If the user exit is used by the markup table, OpenTM2 uses the context information and the translation proposals to identify matches on the segments to be translated.

- is called once when a markup table is loaded. It returns information about the number and the names of context strings used in the Translation Memory, and it controls (based on the availability of context information) whether further context information processing is performed.

user exit entry point EQFGETCONTEXTINFO

- is called before a translated segment is saved in the Translation Memory. It gets the context strings from the user exit and passes them to the Translation Memory.

user exit entry point EQFGETSEGCONTEXT

- is called subsequently for every segment during the analysis of a document and updates the user exit with the context strings from the Translation Memory for the current segment.

user exit entry point EQFUPDATECONTEXT

- is called for every segment and compares and ranks a segment's context information against Translation Memory proposals.

user exit entry point EQFCOMPARECONTEXT

OpenTM2 uses these user exit entry points to support the translation of Lotus Notes forms that contain the Form, Subform, Title, and Subtitle context strings.

EQFGETCONTEXTINFO

Purpose

programming interface calls EQFGETCONTEXTINFO EQFGETCONTEXTINFO

EQFGETCONTEXTINFO is called once when a new markup table is loaded into the Translation Memory. It returns the number of context strings that are used by this markup and the names of these context strings (for example, Panel ID for MRI markup). If a markup table user exit does not support this entry point, or returns an error code, no further context information processing is performed for this markup table (neither *EQFGETSEGCONTEXT*, *EQFUPDATECONTEXT*, nor *EQFCOMPARECONTEXT* is called).

Format

EQFGETCONTEXTINFO(*pusNumOfContextStrings*, *pContextNames***)**

Parameters

- *pusNumOfContextStrings*

The pointer to a USHORT variable receiving the number of context strings that are used by this markup.

- *pContextNames*

The pointer to a UTF16 buffer for the context names. This buffer has a size of MAX_CONTEXT_LEN(4096) characters. The context names are stored as a list of UTF-16 strings, and the list is terminated by 0x0000. Currently the names will not be used. In a later version these names will be used in the translation environment to display the context of a segment.

Return code

The return code indicates whether context information could be returned.

EQFGETSEGCONTEXT

Purpose

programming interface calls EQFGETSEGCONTEXT EQFGETSEGCONTEXT

EQFGETSEGCONTEXT returns the context strings for a given segment and passes them to the Translation Memory functions before a segment is about to be saved in the Translation Memory. This function is used by the editor during the translation. Using the supplied document handle the function can go backward or forward to other segments if necessary (for example, for an MRI markup it is necessary to go back to the segment containing the panel ID).

Format

EQFGETSEGCONTEXT(*pCurSeg*, *pPrevSeg*, *pNextSeg*, *pContextStrings*, *hEditor***)**

Parameters

- *pCurSeg*
The pointer to a zero-terminated UTF-16 string containing the text of the current segment.
- *pPrevSeg*
The pointer to a zero-terminated UTF-16 string that contains the text of the previous segment (NULL, if there is none).
- *pNextSeg*
The pointer to a zero-terminated UTF-16 string that contains the text of the next segment (NULL, if there is none).
- *pContextStrings*
The pointer to a UTF16 buffer for the context strings. This buffer has a size of MAX_CONTEXT_LEN (4096) characters. The context strings are stored as a list of UTF-16 strings, and the list is terminated by 0x0000.
- *hEditor*
The handle of type HANDLE, which is required for the EQFGetNextSeg and EQFGetPrevSeg functions.

Return code

The return code indicates whether context strings could be returned.

EQFUPDATECONTEXT

Purpose

programming interface callsEQFUPDATECONTEXT EQFUPDATECONTEXT

EQFUPDATECONTEXT is called subsequently during the analysis of a document. If the current segment in the Translation Memory contains context information, this function updates the user exit with the context strings for this segment. The retrieved context strings are used to identify exact context matches with the *EQFCOMPARECONTEXT* function.

Format

EQFUPDATECONTEXT(*pSeg, pContextStrings*)

Parameters

- *pSeg*
The pointer to a zero-terminated UTF-16 string containing the text of the current segment.
- *pContextStrings*
The pointer to a UTF16 buffer containing the current context strings and receiving the updated context strings. This buffer has a size of MAX_CONTEXT_LEN(4096) characters. The context strings are stored as a list of UTF-16 strings, and the list is terminated by 0x0000.

Return code

The return code indicates whether context strings could be updated.

EQFCOMPARECONTEXT

Purpose

programming interface callsEQFCOMPARECONTEXT EQFCOMPARECONTEXT

EQFCOMPARECONTEXT is called for every segment that has an exact text match and context information available. The function compares the context strings of a segment against the context strings of a Translation Memory proposal and ranks the match between 0 and 100. 0 means no context match at all, and 100 means an exact context match.

During an analysis only exact text matches *and* exact context matches of a segment lead to automatic substitutions. During a translation, the ranks are used to identify the best translation proposals.

Format

EQFCOMPARECONTEXT(*pContextStrings1, pContextStrings2, pusRanking*)

Parameters

- *pContextStrings1*
The pointer to a buffer containing the context strings of the current segment. The context strings are stored as a list of UTF-16 strings, and the list is terminated by 0x0000.
- *pContextStrings2*
The pointer to a buffer containing the context strings of the proposal. The context strings are

stored as a list of UTF-16 strings, and the list is terminated by 0x0000.

- *pusRanking*
The pointer to the variable receiving the ranking for the context strings.

Return code

The return code indicates whether context information could be compared.

Parser application programming interface

The following functions are internal OpenTM2 parsing functions that are made available to expand the possibilities of user exists. Their main purposes are:

- To access and modify segmented documents on a segment base.
Documents can be loaded, and their segments can be retrieved and modified. Segments can be converted into an SGML tagged format. Code conversions can be done, and some document properties can be retrieved. Modified documents can be saved.
- To access and tokenize markup tables to get information about markup tags and property information.
Markup tables can be loaded and tokenized, and the properties of markup tags can be accessed.

Because these are basically parsing functions, their names start with "Pars". Function names ending with "W" are for Unicode documents, and for markup tables to be used with Unicode documents.

Note that these functions are not called at defined OpenTM2 processing steps (as opposed to the descriptions in [#unresolvedid/gen_user_exit_entry_points](#) and [User exit entry points for context-dependent translations](#)). However, they are well suited to be used in the code of one or more of these entry points. For example, they can be used to create or clean up markup tables. A sample parser that uses these parser API functions can be found in file `parssamp.c` in directory `\eqf\nondde\`.

Further details about these functions, like the definition of data types, can be found in file `eqfpapi.h` in the same directory.

The following sections describe the parser API functions in detail. Where applicable, the parser API functions are enabled for Unicode UTF-16 support. [Unicodesupport for parser API parser](#) [APIUnicode support](#)

ParsInitialize

Purpose

[programming interface calls](#)`ParsInitialize` `ParsInitialize` parser API `ParsInitialize` initializes the parser API environment and creates a parser API handle that is to be used in most of the other parser API functions.

Format

`ParsInitialize(phParser, pszDocPathName)`

Parameters

Type	Parameter	Description
HPARSER	phParser	The pointer to the buffer for the parser API handle.
CHAR	pszDocPathName	The pointer to the zero-terminated document path name.

Return code

Integer of 0, if the environment is successfully initialized, or an error code.

ParsBuildTempName

Purpose

[programming interface calls](#)`ParsBuildTempName` `ParsBuildTempName` parser API `ParsBuildTempName` `ParsBuildTempName` builds a temporary file name based on the fully qualified file name of the source document.

Format

`ParsBuildTempName(pszSourceName, pszTempName)`

Parameters

Type	Parameter	Description
PSZ	pszSourceName	The pointer to the zero-terminated fully qualified file name of the source document. The name serves as the model for the temporary file name.
PSZ	pszTempName	The pointer to the zero-terminated temporary file name. The buffer for the file name should have a size of 128 bytes or more.

Return code

Integer of 0, if the file name is successfully built, or an error code.

ParsLoadSegFile

Purpose

programming interface calls `ParsLoadSegFile` `ParsLoadSegFile` parser API `ParsLoadSegFile`
`ParsLoadSegFile` loads a segmented file into memory.

Format

`ParsLoadSegFile(hParser, pszFileName, phSegFile)`

Parameters

Type	Parameter	Description
HPARSER	hParser	The parser API handle, created by the <code>ParsInitialize</code> function.
CHAR	pszFileName	The pointer to the zero-terminated fully qualified file name of the document to be loaded into memory.
HPARSSEGFILE	phSegFile	The pointer to the buffer in memory that receives the segmented file.

Return code

Integer of 0, if the file is successfully loaded, or an error code.

ParsGetSegNum

Purpose

programming interface calls `ParsGetSegNum` `ParsGetSegNum` parser API `ParsGetSegNum`
`ParsGetSegNum` returns the number of segments of the segmented file loaded into memory.

Format

`ParsGetSegNum(hSegFile, plSegCount)`

Parameters

Type	Parameter	Description
HPARSSEGFILE	phSegFile	The handle of the segmented file in memory.
LONG	plSegCount	The pointer to the buffer that receives the number of segments.

Return code

Integer of 0, if the number is successfully retrieved, or an error code.

ParsGetSeg

Purpose

programming interface calls `ParsGetSeg` `ParsGetSeg` parser API `ParsGetSeg` `ParsGetSeg` gets a segment from the segmented file loaded into memory. If the segment in Unicode format, use [ParsGetSegW](#).

Format**ParsGetSeg(hSegFile, lSegNum, pSeg)****Parameters**

Type	Parameter	Description
HPARSSEGFILE	hSegFile	The handle of the segmented file in memory.
LONG	lSegNum	The number of the segment to get.
PPARSSEGMENT	pSeg	The pointer to the buffer that receives the segment data.

Return code

Integer of 0, if the segment is successfully retrieved, or an error code.

ParsGetSegW**Purpose**programming interface callsParsGetSegW ParsGetSegW parser APIParsGetSegW ParsGetSegW gets a segment from the segmented file loaded into memory. If the segment not in Unicode format, use [ParsGetSeg](#).**Format****ParsGetSegW(hSegFile, lSegNum, pSeg)****Parameters**

Type	Parameter	Description
HPARSSEGFILE	hSegFile	The handle of the segmented file in memory.
LONG	lSegNum	The number of the segment to get.
PPARSSEGMENTW	pSeg	The pointer to the buffer that receives the segment data.

Return code

Integer of 0, if the segment is successfully retrieved, or an error code.

ParsUpdateSeg**Purpose**programming interface callsParsUpdateSeg ParsUpdateSeg parser APIParsUpdateSeg ParsUpdateSeg updates a segment of the segmented file loaded into memory. If the segment is in Unicode format, use [ParsUpdateSegW](#).**Format****ParsUpdateSeg(hSegFile, lSegNum, pSeg)****Parameters**

Type	Parameter	Description
HPARSSEGFILE	hSegFile	The handle of the segmented file in memory.
LONG	lSegNum	The number of the segment to update.
PPARSSEGMENT	pSeg	The pointer to the buffer that holds the new segment data.

Return code

Integer of 0, if the segment is successfully updated, or an error code.

ParsUpdateSegW**Purpose**programming interface callsParsUpdateSegW ParsUpdateSegW parser APIParsUpdateSegW ParsUpdateSegW updates a segment of the segmented file loaded into memory. If the segment is not in Unicode format, use [ParsUpdateSeg](#).

Format

ParsUpdateSegW(hSegFile, lSegNum, pSeg)

Parameters

Type	Parameter	Description
HPARSSEGFILE	hSegFile	The handle of the segmented file in memory.
LONG	lSegNum	The number of the segment to update.
PPARSSEGMENTW	pSeg	The pointer to the buffer that holds the new segment data.

Return code

Integer of 0, if the segment is successfully updated, or an error code.

ParsWriteSegFile**Purpose**

programming interface calls **ParsWriteSegFile** **ParsWriteSegFile** parser API **ParsWriteSegFile** **ParsWriteSegFile** writes the segmented file in memory to an external file.

Format

ParsWriteSegFile(hSegFile, pszFileName)

Parameters

Type	Parameter	Description
HPARSSEGFILE	hSegFile	The handle of the segmented file in memory.
CHAR	pszFileName	The pointer to the zero-terminated fully qualified file name of the document.

Return code

Integer of 0, if the file is successfully written, or an error code.

ParsMakeSGMLSegment**Purpose**

programming interface calls **ParsMakeSGMLSegment** **ParsMakeSGMLSegment** parser API **ParsMakeSGMLSegment** **ParsMakeSGMLSegment** builds an SGML tagged segment as used in segmented files. If the segment is in Unicode format, use [ParsMakeSGMLSegmentW](#).

Format

ParsMakeSGMLSegment(hParser, pSegment, pszBuffer, iBufferSize, fSourceFile)

Parameters

Type	Parameter	Description
HPARSER	hParser	The parser API handle, created by the <i>ParsInitialize</i> function.
PPARSSEGMENT	pSegment	The pointer to the buffer that holds the segment data.
CHAR	pszBuffer	The pointer to the buffer that receives the zero-terminated SGML-tagged segment. The buffer size for the segment should be at least twice the maximum segment size.
INT	iBufferSize	The size of <i>pszBuffer</i> .
BOOL	fSourceFile	<ul style="list-style-type: none"> • TRUE Create SGML for a segmented source file. • FALSE Create SGML for a segmented target file.

Return code

Integer of 0, if the segment is successfully built, or an error code.

ParsMakeSGMLSegmentW**Purpose**

programming interface calls `ParsMakeSGMLSegmentW` `ParsMakeSGMLSegmentW` parser API `ParsMakeSGMLSegmentW` `ParsMakeSGMLSegmentW` builds an SGML tagged segment as used in segmented files. If the segment is not in Unicode format, use [ParsMakeSGMLSegment](#).

Format

`ParsMakeSGMLSegmentW(hParser, pSegment, pszBuffer, iBufferSize, fSourceFile)`

Parameters

Type	Parameter	Description
HPARSER	hParser	The parser API handle, created by the <code>ParsInitialize</code> function.
PPARSSEGMENTW	pSegment	The pointer to the buffer that holds the segment data.
WCHAR*	pszBuffer	The pointer to the buffer that receives the zero-terminated SGML-tagged segment (in Unicode UTF-16 format). The buffer size for the segment should be at least twice the maximum segment size.
INT	iBufferSize	The size of <code>pszBuffer</code> .
BOOL	fSourceFile	<ul style="list-style-type: none"> • TRUE Create SGML for a segmented source file. • FALSE Create SGML for a segmented target file.

Return code

Integer of 0, if the segment is successfully built, or an error code.

ParsConvert**Purpose**

programming interface calls `ParsConvert` `ParsConvert` parser API `ParsConvert` `ParsConvert` performs an in-place conversion from ASCII to ANSI, or vice versa.

Format

`ParsConvert(hParser, Conversion, pszData, usLen)`

Parameters

Type	Parameter	Description
HPARSER	hParser	The parser API handle, created by the <code>ParsInitialize</code> function.
PARSCONVERSION	Conversion	The conversion mode: <ul style="list-style-type: none"> • ASCIItoANSI • ANSItoASCII
CHAR	pszData	The pointer to the zero-terminated data to be converted.
USHORT	usLen	The length of the data to convert.

Return code

Integer of 0, if the conversion is successful, or an error code.

ParsGetDocName

Purpose

programming interface calls `ParsGetDocName` parser API `ParsGetDocName`.
`ParsGetDocName` returns the long document name.

Format

`ParsGetDocName(hParser, pszDocName)`

Parameters

Type	Parameter	Description
HPARSER	hParser	The parser API handle, created by the <code>ParsInitialize</code> function.
CHAR	pszDocName	The pointer to the buffer that receives the zero-terminated long document name. The size of the buffer should be 256 bytes.

Return code

Integer of 0, if the document name is successfully returned, or an error code.

ParsGetDocLang

Purpose

programming interface calls `ParsGetDocLang` parser API `ParsGetDocLang`.
`ParsGetDocLang` returns the language settings of the current document.

Format

`ParsGetDocLang(hParser, pszSourceLang, pszTargetLang)`

Parameters

Type	Parameter	Description
HPARSER	hParser	The parser API handle, created by the <code>ParsInitialize</code> function.
CHAR	pszSourceLang	The pointer to the buffer that receives the zero-terminated source language, or NULL. The buffer size should be 40 bytes or more.
CHAR	pszTargetLang	The pointer to the buffer that receives the zero-terminated target language, or NULL. The buffer size should be 40 bytes or more.

Return code

Integer of 0, if the language setting are successfully returned, or an error code.

ParsSplitSeg

Purpose

programming interface calls `ParsSplitSeg` parser API `ParsSplitSeg`.
`ParsSplitSeg` splits text data into segments by using OpenTM2's morphological functions. The function looks for segment breaks in the supplied data by applying the morphology for the document source language. The segment breaks are returned as a list of segment breaks. This list is a list of offsets of segment breaks within the data. The last element in this list is zero.

If the buffer for this list is too small, the function returns an error and the first element of the list contains the required size of the list (in number of list elements).

If the text data is in Unicode format, use [ParsSplitSegW](#).

Format

`ParsSplitSeg(hParser, pszData, usDataLength, pusSegBreaks, usElements)`

Parameters

Type	Parameter	Description
HPARSER	hParser	The parser API handle, created by the <i>ParsInitialize</i> function.
CHAR	pszData	The pointer to the zero-terminated text data that is to be split into segments.
USHORT	usDataLength	The length of the text data, as number of characters.
USHORT	pusSegBreaks	The pointer to the buffer that receives the list of segment breaks.
USHORT	usElements	The size of the buffer that receives the list of segment breaks, in number of list elements.

Return code

Integer of 0, if the segment is successfully split, or an error code.

ParsSplitSegW

Purpose

programming interface calls **ParsSplitSegW** **ParsSplitSegW** parser API **ParsSplitSegW**
ParsSplitSegW splits text data into segments by using OpenTM2's morphological functions. The function looks for segment breaks in the supplied data by applying the morphology for the document source language. The segment breaks are returned as a list of segment breaks. This list is a list of offsets of segment breaks within the data. The last element in this list is zero.

If the buffer for this list is too small, the function returns an error and the first element of the list contains the required size of the list (in number of list elements).

If the text data is not in Unicode format, use [ParsSplitSeg](#).

Format

ParsSplitSegW(hParser, pszData, usDataLength, pusSegBreaks, usElements)

Parameters

Type	Parameter	Description
HPARSER	hParser	The parser API handle, created by the <i>ParsInitialize</i> function.
WCHAR*	pszData	The pointer to the zero-terminated text data (in Unicode UTF-16 format) that is to be split into segments.
USHORT	usDataLength	The length of the text data, as number of UTF-16 characters.
USHORT	pusSegBreaks	The pointer to the buffer that receives the list of segment breaks.
USHORT	usElements	The size of the buffer that receives the list of segment breaks, in number of list elements.

Return code

Integer of 0, if the segment is successfully split, or an error code.

ParsFreeSegFile

Purpose

programming interface calls **ParsFreeSegFile** **ParsFreeSegFile** parser API **ParsFreeSegFile**
ParsFreeSegFile frees a segmented file from memory.

Format

ParsFreeSegFile(hSegFile)

Parameters

Type	Parameter	Description
HPARSSEGFILE	hSegFile	The handle of the segmented file in memory.

Return code

Integer of 0, if the memory is successfully freed, or an error code.

ParsLoadMarkup

Purpose

programming interface calls `ParsLoadMarkup` `ParsLoadMarkup` parser API `ParsLoadMarkup`. `ParsLoadMarkup` loads a markup table into memory for usage with the `ParsTokenize` or `ParsTokenizeW` function. The markup table is loaded from the `\eqf\table` directory.

Format

`ParsLoadMarkup(hParser, phMarkup, pszMarkup)`

Parameters

Type	Parameter	Description
HPARSER	hParser	The parser API handle, created by the <code>ParsInitialize</code> function.
HPARSMARKUP*	phMarkup	The pointer to the buffer in memory that receives the markup handle.
CHAR	pszMarkup	The pointer to the zero-terminated markup table name (without path and extension, for example, <code>EQFANSI</code>).

Return code

Integer of 0, if the markup table is successfully loaded, or an error code.

ParsTokenize

Purpose

programming interface calls `ParsTokenize` `ParsTokenize` parser API `ParsTokenize`. `ParsTokenize` looks for tags in the supplied text area of the markup table loaded into memory. The result is a tag token list that can be processed by the `ParsGetNextToken` function.

If the supplied text area is in Unicode format, use [ParsTokenizeW](#).

Format

`ParsTokenize(hMarkup, pszData)`

Parameters

Type	Parameter	Description
HPARSMARKUP	hMarkup	The markup handle, created by the <code>ParsLoadMarkup</code> function.
CHAR*	pszData	The pointer to the zero-terminated text area that is to be tokenized.

Return code

Integer of 0, if the markup table is successfully tokenized, or an error code.

ParsTokenizeW

Purpose

programming interface calls `ParsTokenizeW` `ParsTokenizeW` parser API `ParsTokenizeW`. `ParsTokenizeW` looks for tags in the supplied text area of the markup table loaded into memory. The result is a tag token list that can be processed by the `ParsGetNextToken` function. If the

supplied text area is not in Unicode format, use [ParsTokenize](#).

Format

[ParsTokenizeW\(hMarkup, pszData\)](#)

Parameters

Type	Parameter	Description
HPARSMARKUP	hMarkup	The markup handle, created by the <i>ParsLoadMarkup</i> function.
WCHAR*	pszData	The pointer to the zero-terminated Unicode text area that is to be tokenized.

Return code

Integer of 0, if the markup table is successfully tokenized, or an error code.

ParsGetNextToken

Purpose

programming interface calls [ParsGetNextToken](#) [ParsGetNextToken](#) parser API [ParsGetNextToken](#). [ParsGetNextToken](#) returns the next token from the token list created by the [ParsTokenize](#) and [ParsTokenizeW](#) functions. At the end of the token list a token with a token ID of PARSTOKEN_ENDOFLIST is returned. [The PARSTOKEN structure](#) describes the token structure in detail.

Format

[ParsGetNextToken\(hMarkup, pToken\)](#)

Parameters

Type	Parameter	Description
HPARSMARKUP	hMarkup	The markup handle, created by the <i>ParsLoadMarkup</i> function.
PPARSTOKEN	pToken	The pointer to a PARSTOKEN structure (see ...) that receives the data of the token.

Return code

Integer of 0, if the next token is returned, or an error code.

[The PARSTOKEN structure](#)

This structure holds the token information of a token that is returned by the [ParsGetNextToken](#) function.

Type	Name	Usage
INT	iTokenID	<p>The token ID of the token returned. The token ID represents the position of the tag in the markup table.</p> <ul style="list-style-type: none">• A token ID of PARSTOKEN_ENDOFLIST represents the end of the tag token list.• A token ID of PARSTOKEN_TEXT (text token) represents text which is not recognized as a tag.
INT	iStart	The start position (in characters, not bytes) of the token in the text area (see ... parameter <i>pszData</i> of the ParsTokenize or ParsTokenizeW function).

Type	Name	Usage
INT	iLength	The length of the token (in number of characters, not bytes).
USHORT	usFixedID	A fixed token ID, or <code>NULL</code> if none is specified for the tag in the markup table.
USHORT	usAddInfo	Additional tag information, or <code>NULL</code> if none is specified for the tag in the markup table.
USHORT	usClassID	A Class ID, or <code>NULL</code> if none is specified for the tag in the markup table.

ParsFreeMarkup

Purpose

programming interface calls `ParsFreeMarkup` `ParsFreeMarkup` parser API `ParsFreeMarkup`
`ParsFreeMarkup` frees a markup table loaded with the `ParsLoadMarkup` function from memory.

Format

`ParsFreeMarkup(hMarkup)`

Parameters

Type	Parameter	Description
HPARSMARKUP	hMarkup	The markup handle, created by the <code>ParsLoadMarkup</code> function.

Return code

Integer of 0, if the markup table is freed from memory, or an error code.

ParsTerminate

Purpose

programming interface calls `ParsTerminate` `ParsTerminate` parser API `ParsTerminate`
`ParsTerminate` terminates the parser API environment.

Format

`ParsTerminate(hParser)`

Parameters

Type	Parameter	Description
HPARSER	hParser	The parser API handle, created by the <code>ParsInitialize</code> function.

Return code

Integer of 0, if the environment is successfully terminated, or an error code.

Appendices

Overview of the OpenTM2 menus

The following table shows which tasks can be selected for which OpenTM2 components.

Note that for every list window you can open a list-specific menu where you can select the same commands as from the OpenTM2 menu bar. You open this menu by pressing the right mouse button. The menu shows only the selection of tasks that can be started from the list window for which you opened the menu.

Documents	Folders	<i>Translation Memory databases</i>	Dictionaries	Markup tables
Open	Open	Open	Open	—
—	New...	New...	New...	—
Properties...	Properties...	Properties...	Properties...	—
Properties Summary	Properties Summary	Properties Summary	Properties Summary	—
Delete	Delete	Delete	Delete	Delete
—	Rename...	Rename...	Rename...	—
—	—	Merge...	—	—
—	—	Organize	Organize	—
Analyze...	Analyze...	—	—	—
—	—	—	Print...	—
Export...	Export...	Export...	Export...	Export...
Import...	Import...	Import...	Import...	Import...
Count words...	Count words...	—	—	—
Create Counting Report...	Create Counting Report...	—	—	—
—	Delete exported folder...	—	—	—
Global find and replace...	Global find and replace...	—	—	—
Select all	—	—	—	—
Deselect all	—	—	—	—
—	—	Connect shared resources...	Connect shared resources...	—

Editor functions reference

To assign a function to a key or a key combination, click **Keys...** in the **Options** menu, select the function and assign it to a key.

To call an editor function that is assigned to a key combination, click **Commands...** in the **Options** menu, select the command to be executed, and click **Run**.

On the following pages, you find the complete list of the functions available in the **Translation Environment**. They are grouped as follows:

- Editor functions inside the "Translation" window
- Editor functions outside the "Translation" window

The tables shown provide a column in which you can enter the key that you assigned to the functions. Note that the keys Ctrl+A to Ctrl+Z and Ctrl+a to Ctrl+z are reserved keys and cannot be assigned.

Editor functions inside the Translation window

Cursor movement in window

Bottom	Cursor	Ctrl+End	
Top	Cursor	Ctrl+Home	
Cursor down		Down	
Cursor up		Up	
Cursor left		Left	
Cursor right		Right	
Insert shift-in/shift-out character in file			
Insert shift-in/shift-out character in segment			
Delete shift-in/shift-out character in file			
Delete shift-in/shift-			

out character in segment			
Start of line	Cursor	Home	
End of line	Cursor	End	
Start of segment	Cursor	Alt+Home	
End of segment	Cursor	Alt+End	
Query (active) line	Cursor		
Go to line dialog	Cursor		
Go to segment dialog	Cursor		
Special go to dialog	Cursor		
Next line		Enter	
Next word		Ctrl+Right	
Previous word		Ctrl+Left	
Tab backward		Backtab	
Tab forward		Tab	
Go to active segment	Translate		
Go to changed segment			
Go to newly translated segment			
Go to unchanged segment			
Go to bookmark	Translate		

Scrolling

Function	Menu	Default key	My key
Scroll down		Shift+F4	
Scroll up		Shift+F3	
Scroll page down		PageDown	
Scroll page up		PageUp	
Scroll left		Shift+F5	
Scroll right		Shift+F2	

Editing

Function	Menu	Default key	My key
Delete character or marked block		Delete	
Delete character and move cursor left		Backspace	
Delete until tag			
Delete line			
Insert line			
Join line(s)	Edit		
Split line	Edit		
Truncate line		Ctrl+Delete	
Delete word			
Mark word			
Compress current segment			
Truncate segment		Alt+Delete	
Delete segment			
Join segments	Translate		
Split joined segments	Translate		
Mark segment			
Next untranslated segment	Translate		
Reflow segment	Edit		
Spellcheck segment	Spellcheck		

Function	Menu	Default key	My key
Translate segment	Translate	Ctrl+Enter	
Untranslate segment	Translate		
Copy block			
Copy proposal block	Edit	Ctrl+letter of proposal	
Delete block (Clear)	Edit		
Find block	Cursor		
Mark block			
Move block			
Unmark block	Edit		
Find (and change...)	Edit		
Undo	Edit	Alt+Backspace	
Set bookmark	Translate		
Clear bookmark	Translate		

Modes of editing

Function	Menu	Default key	My key
Line wrap (toggle)	Edit		
Insert toggle		Insert	
Toggle first character of dictionary proposal			
Add an abbreviation	Translate		
Edit abbreviations	Translate		
Postediting	Translate		
Show translation	Translate		
Automatic substitution	Translate		
Automatic spellchecking	Spellcheck		
Spellcheck file	Spellcheck		
Spellcheck segment	Spellcheck		
Next misspelled word	Spellcheck		
Next misspelled word	Spellcheck		
Next updated word			

Changing the presentation of control tags

Function	Menu	Default key	My key
Compact (document)	Style		
Compact+1 (document)	Style		
Hide tags	Style		
Protect tags	Style		
Shrink tags (in segment)	Style		
Unprotect tags	Style		

Document overall functions

Function	Menu	Default key	My key
File (End_Save)	File	F4	
Open	File		
Quit	File	F3	
Print (document)	File		
Save	File	F2	
Next (document in the ring)	File		

Function	Menu	Default key	My key
Previous document			

Switching to other windows

Function	Menu	Default key	My key
"Set Colors" window	Options		
"Set Fonts" window	Options		
"Run Command" window	Options		
"Assign Keys" window	Options		
"Customize Translation Functions" window	Options		
"Sentence Lookup" window	Options		
"Translator's Note" window			

Editor functions outside the Translation window

Using the clipboard

Function	Menu	Default key	My key
Copy (to clipboard)	Edit	Ctrl+Insert	
Cut (to clipboard)	Edit	Shift+Delete	
Paste (from clipboard)	Edit	Shift+Insert	

Switching to other windows

Function	Menu	Default key	My key
"Dictionary" window	File		
"Original" window	File		
"Source of proposal" window	File		
"Translation Memory" window	File	F8	
"Translation" window	File		
Look up a term	Translate		
Edit a term	Translate		

Functions in the Translation Memory window

Function	Menu	Default key	My key
Scroll proposal down		Ctrl+Down	
Scroll proposal up		Ctrl+Up	
Display (also) fuzzy matches			
Display all exact translation proposals			

Functions in the Dictionary window

Function	Menu	Default key	My key
Scroll dictionary down		Alt+Down	

Function	Menu	Default key	My key
Scroll dictionary up		Alt+Up	

Format files for printing a dictionary

To print dictionary data, you must first define the layout of your print output in format files that you edit with a text editor. In the format print file you define the entry fields to be printed and how to arrange them in the printout. For general information on printing, see [Printing a dictionary](#).

Sample format files

OpenTM2 provides several sample format files. The following contains an introduction to format file statements.

The print program assumes a page size with a default of 72 lines per page. To modify this layout you can insert printer-specific characters in the format files according to the syntax provided.

FORMAT1.FRM

This format file prints each headword and its translation. It uses the following format statements to achieve this:

Part of FORMAT1.FRM

```
<entryprt>
=====
<var name=Headword>
<var name=Translation>
</entryprt>
```

Between `<entryprt>` and `</entryprt>` you specify which entry fields you want to be printed for each entry. The names following `var name=` must be the same names that are defined as entry field names in the dictionary you want to print. To see which entry field names are in a dictionary, select **Properties...** from the **File** menu when the "Dictionary List" window is active and the respective dictionary selected.

`<var name=Translation>` means that the translation for the headword is printed. If there is more than one translation, only the first translation is printed.

Nontagged information that you type between the `<entryprt>` and `</entryprt>` tags is printed for each iteration. In the example above, the `=====` is used as a separator line between the entries on your printout.

The following is an example of what your dictionary printout would look like. Assume that the dictionary is English-German and the sample format FORMAT1.FRM has been used for formatting. Two entries (rule and run) are extracted.

Sample printout

```
=====
rule
regieren
=====
run
laufen
```

FORMAT2.FRM

This format file prints the templates for each entry without formatting. A template is all entry field information on all levels (entry level, homonym level, sense level, and target level) relating to one specific translation of a headword. The following is an illustration of a dictionary entry.

FORMAT2.FRM looks as follows:

Part of FORMAT2.FRM

```
<entryprt>
=====
<var name =Headword>
<repeat name=Headword>
<var name='Part of Speech'>
<var name=Translation>
<var name=Abbrev./Fullform>
</repeat>
</entryprt>
```

<var name=Headword> means that the headword is printed. <repeat...> means that the Part of Speech, the Translation, and the Abbrev./Fullform entry fields are printed until a different headword is found. If the fields for Part of Speech, Translation, or Abbrev./Fullform in the dictionary are not filled, an empty line is printed.

The following sample shows how your dictionary printout would look. Assume that the dictionary is English-German and the sample format FORMAT2.FRM has been used for formatting. Two entries (rule and run) are extracted.

Sample printout

```
=====
rule
verb
regieren

rule
noun
Regel

=====
run
verb
laufen

run
noun
Lauf
```

FORMAT3.FRM

This format file prints all templates for each entry.

It is designed to be formatted with BookMaster^(R) outside **OpenTM2** before you print it. If you print the file without BookMaster^(R), you see a mixture of dictionary entry data and BookMaster^(R) tags. Formatting with BookMaster^(R) produces a printout resembling entries in printed dictionaries.

The tags used in this format file are described in [Tags for defining the format of a dictionary printout](#). You must use them in the same sequence as listed there.

FORMAT4.FRM

This format file prints a dictionary extract with the following structure:

- A front page showing the title of the printed dictionary, the date, and the author.
- The major part containing the dictionary entries with the fields specified by you. The sample contains the fields Translation, Part of Speech, Subject Code, and Context.
- A header section for each page with the name of the dictionary and the first headword on each page.
- The bottom of each page displaying the current page number.

FORMAT4.FRM contains comments on most of the tags used. It is recommended that you read these comments to understand the results the tags produced on the printout.

If you want to see the results of such a printout, print dictionary DICTPRT, which is contained in sample folder SAMPLPRT.

Note:

You must first import SAMPLPRT into **OpenTM2**.

Defining your own format file

To print a dictionary or an extract from it, you can use one of the format files provided by **OpenTM2**. You find them on the drive where you installed **OpenTM2** in the subdirectory \eqf\prtform.

These format files provide a predefined layout for printing dictionary information. They serve as sample files. You must replace all the generic information they contain with your personal data, such as your address, and your name. It is useful to select a format file that contains entry fields that match those of the dictionary you want to print.

To get an idea of how the printout will turn out when using one of these format files, read the explanation that is given after the <comment> tag in each file. However, if you want to print a dictionary with a layout other than those provided, you can define your own format file. You do this as follows:

1. Copy one of the format files provided or create a new file. It must be in ASCII format.
2. Give it the extension `frm` (for example, `myform.frm`).
3. Edit this file with a text editor of your choice. You can use any of the tags given in **...** to create an individual layout.
4. Place this file in the subdirectory `\eqf\prtform` on the drive where you installed **OpenTM2**.
5. Use the new file name as format file name in the "Print Dictionary" window.

The selected dictionary is printed on the printer that is defined as LPT1 in your operating system (default setting).

The following is an example of an individual format file and how it looks on the printout.

```
<comment>
This is my private format file.
It prints entries from dictionary MYDICT in the following way.
The front page gives administrative information about the
printout, such as title, date, and my name.
Then two entries of MYDICT follow.
These entries contain the fields I specified:
Part of Speech, Translation, Synonym, and Context.
Each page has the dictionary name MYDICT in its header section.
The page number is printed at the bottom of each page.
</comment>
<set sysname=$page_length value='61'>
<set sysname=$page_eject value=' '>
<dictfront>

Title: Dictionary <var sysname=$Dictname>

Date: <var sysname=$Date format=dd.mm.yyyy>

Author: Betty Miller
</dictfront>
<pagehead>
<var sysname=$Dictname $Min=45 $Right>
<var name=headword $first_on_page>
=====
</pagehead>
<entryprt>
```

The term "<var name=Headword>" has the following meaning:

```
<repeat name=Headword>
  Part of Speech: <var name='Part of Speech'>
<repeat name='Part of Speech'>
  Translation: <var name=Translation>
  Synonym: <var name=Synonym>
  Context: <var name=Context>

</repeat>
</repeat>
<entryprt>
<pagefoot>
=====
```

- <var sysname=\$Page_No> -

If you used this format file when printing dictionary MYDICT, you would get a printout with the following information on its title page:

Title: Dictionary MYDICT

Date: 04/01/94

Author: Betty Miller

After the title page, the entries found are printed in the following way:
MYDICT

assist

The term "assist" has the following meaning:

Part of Speech: verb

Translation: helfen

Synonym:

Context:

Translation: assistieren

Synonym:

Context: to assist sb

The term "access" has the following meaning/s:

Part of Speech: noun

Translation: Zugriff

Synonym:

Context:

Translation: Zutritt

Synonym:

Context:

Part of Speech: verb

Translation: zugreifen

Synonym:

Context:

=====

- 1 -

Tags for defining the format of a dictionary printout

Each data element to be printed is defined between a start and an end tag (except for `<set>` and `<var>`). The end tag always starts with "`</`". If you use any multiword terms as entry-field names, for example, Part of Speech, you must enclose them in single quotes (for example: `<var name='Part of Speech'>`). You may type these tags in either uppercase or in lowercase letters. Some of the tags have attributes (`<set...>`, `<repeat...>`, `<var...>`).

You can control the layout of your printout by placing the variables at the appropriate position. The printing of a text or a data element starts where the opening pointed bracket (`<`) has been placed.

Start tag, end tag	Meaning
<code><description>, </description></code>	Here you can describe what your printout contains. These tags are optional and have no effect on the layout of your printout. This description is shown in the "Print Dictionary" window.
<code><dictfront>, </dictfront></code>	Contains administrative information related to the printed dictionary, such as the name of the dictionary and the date of printing. This information is printed once at the beginning of your printout before any entries are printed. These tags are optional.
<code><pagehead>, </pagehead></code>	Here you can specify the information that you want to be printed on the top of each printed page. These tags are optional.
<code><entryprt>, </entryprt></code>	Here specify the structure and contents of the entries you want to be printed. These tags are mandatory.
<code><pagefoot>, </pagefoot></code>	Here you can specify the information that you want to be printed at the bottom of each printed page. These tags are optional.
<code><dictback>, </dictback></code>	Contains information related to the printed dictionary. This information is printed at the end of your printout after all entries have been printed. These tags are optional.

Within the format files the tags must follow the same sequence as specified here. To modify the layout of your printout, you can use the following additional tags:

Start tag, end tag	Meaning
<comment>, </comment>	Here you can specify information that you do not want to appear on your printout. These tags are optional. They can occur anywhere within a format file, but not between another start and end tag.
<set...>	This tag lets you specify the length of a line and the length of a page of your printout. It can occur anywhere within a format file, but not between another start and end tag. For more information, see
<repeat...>, </repeat>	The information specified here is repeated until all information belonging to the specified entry field is printed. These tags are optional and can only occur between <entryprt> and </entryprt>. For more information, see
<var...>	This tag can have the attribute "name=". It can occur only between the <pagehead>, <pagefoot>, and <entryprt> tags and their end tags. If it has the attribute "sysname=\$..." it can occur only between the <dictfront>, <pagehead>, <entryprt>, <pagefoot>, and <dictback> tags and their corresponding end tags. For more information, see The <var...> attributes .

The <set...> attributes

- **sysname=\$line_length**
Specifies the length of a single line on your printout. If you do not use this attribute, 80 characters per line are printed. This attribute must be used together with the **value** attribute, for example: <set sysname=\$line_length value='70'>.
- **sysname=\$page_length**
Specifies the number of lines that are to be printed per page. If you do not use this attribute, 72 lines per page are printed. If your printer prints less than 72 lines per page, you must specify the correct number here. This attribute must be used together with the **value** attribute, for example: <set sysname=\$page_length value='62'>
- **sysname=\$page_no**
Specifies which page number is to be printed on the first page of your printout. If you do not use this attribute, the starting page number is 1. This attribute must be used together with the **value** attribute, for example: <set sysname=\$page_no value='7'>
- **sysname=\$page_eject**
Specifies a character string that is inserted when the printer is supposed to make a page eject. You must specify the value that is used by your printer, otherwise no page eject is made. Refer to your printer manual for the page eject character that is used by your printer (for example the ♀ character = ASCII character 12). If you do not use this attribute, no character is used. An automatic page eject is done within the <dictfront> and <dictback> section. This attribute must be used together with the **value** attribute, for example: <set sysname=\$page_eject value='♀'>.
- **value=xxx**
This attribute is mandatory. It must be used in combination with the attributes listed here for the <set...> tag. You use it to specify the line length, the page length, the page number, and the page eject string.

The <repeat...> attributes

- **name=xxx**
Specifies that the information between <repeat...> and </repeat> is repeated as long as the contents of the xxx entry field is the same while processing the templates of an entry.
- **level=entry|hom|sense|target**
Specifies that the information between <repeat...> and </repeat> is repeated as long as the contents of all fields of the specified level is the same. For a description of the individual levels, refer to [Dictionary entry structure](#).

The <var...> attributes

- **name=xxx**

Specifies which entry field (xxx) of a dictionary entry is to be printed. It can be used together with one of the following attributes:

- **\$first_on_page**

Prints the content of the specified entry field when a page is started. This lets you put the headword in the pagehead section. For example:

```
<var name=headword $first_on_page>
```

- **\$last_on_page**

Prints the contents of the specified entry field when a page is finished. This lets you put the headword in the pagefoot section. For example:

```
<var name=headword $last_on_page>
```

- **\$same_entry_again**

Makes a reset within one entry going from the end of one template to the beginning of the same template again. This lets you process the same template again, printing different information. For example:

```
<var name=headword $same_entry_again>
```

- **\$no_display**

Does not print the contents of the specified entry field. You can use it in combination with **\$same_entry_again**

when only a reset is needed, but no information is to be printed. For example:

```
<var name=headword $no_display>
```

- **sysname=\$filename**

Specifies the name of the file your printout is to be written to.

- **sysname=\$dictname**

Specifies the name of the dictionary that you selected for printing.

- **sysname=\$date**

Prints the current date at the specified position.

- **sysname=\$time**

Prints the current time at the specified position.

- **format=date/time**

Specifies the format of the date and time information. This attribute is mandatory for **sysname=\$date** and **sysname=\$time**. For example: <var sysname=\$date format='dd.mm.yy'> or <var sysname=\$time format='hh:mm:ss'>

- **sysname=\$page_no**

Prints the current page number.

- **sysname=\$page_eject**

Makes a page eject at this position.

- **\$max=n**

Specifies the maximum number of characters that is to be printed from the information in the specified entry field. If the information exceeds the specified number of characters, the rest is not printed.

- **\$min=n**

Specifies the minimum number of characters to be printed from the information in the specified entry field. This tag lets you define a minimum amount of space for an entry field. You may want to use it if you wish to print the contents of entry fields in two columns. If the information in the entry field is less than the specified number of characters, the rest is filled with blank characters. If both **\$max** and **\$min** are specified, the value for **\$min** must be lower or equal to the value of **\$max**.

- **\$left**

The printed text of the specified entry field is justified at the left edge. This attribute only works in combination with the **\$min** attribute. For example: <var sysname=\$dictname \$min=45 \$left>.

- **\$right**

The printed text of the specified entry field is justified at the right edge. This attribute only works in combination with the **\$min** attribute. For example: <var sysname=\$dictname \$min=45 \$right>.

Displaying markup information for Word documents

The following table contains all markup tags that are used for Microsoft^(R)Word for Windows^(R). The first column gives the tag name, the second column indicates whether the tag has an end tag or not, and the third column gives a brief description of the tag.

It may be necessary to insert highlighting tags like bold, italic, or underline if the sentence changes its structure in the target language. If you insert a highlighting tag, the corresponding end tag must be inserted, too. The first table lists the tags that reflect such highlighting that may be changed.

Most of the used markup tags for Microsoft^(R) Word for Windows^(R) have a start tag and an end tag. The end tag has the same name as the start tag, with a slash in front of the tag name. When the end tag is different from the start tag, it is given in addition to the start tag.

Markup data that can be changed

The following table lists markup data that may be deleted, inserted, or moved depending on the structure in the target language. These tags are parts of a segment that is to be translated.

Tag name	End tag	Description
[<i>-</i>]	no	Required hyphen.
[blank]	no	Required blank.
[bold]	yes	The text is printed in bold.
[caps]	yes	The text is printed as capital letters.
[color: <i>code</i>] Do not change the variables, because they are calculated automatically by [<i>color</i>]	yes	The text is printed colored; the <i>code</i> is the color used.
[condensed: <i>space</i>] Do not change the variables, because they are calculated automatically by [<i>condensed</i>]	yes	The text is printed in condensed form, the respective space between the characters is coded with <i>space</i> .
[extended: <i>space</i>] Do not change the variables, because they are calculated automatically by Changed or added markup data when dealing with documents originated in Changed or added markup data when dealing with documents originated in	yes	The text is printed in expanded form, the respective space between the characters is coded with <i>space</i> .
[font: <i>name</i>] Do not change the variables, because they are calculated automatically by [<i>font</i>]	yes	The text is printed in a special font; the font name is given with <i>name</i> .
[fontsize: <i>size</i>] Do not change the variables, because they are calculated automatically by [<i>fontsize</i>]	yes	The text is printed in a special size; the fontsize is coded with <i>size</i> .
[italic]	yes	The text is printed in italic.
[lid: <i>lid code</i>] Do not change the variables, because they are calculated automatically by [<i>lid</i>]	yes	The text has a specific language reference; refer to your Microsoft ^(R) Word for Windows ^(R) documentation for the meaning.
[line break]	no	Soft line feed.
[page]	no	Start of a new page.
[rev mark]	yes	The text is revised.
[rev strike]	yes	The text is revised by strikethrough.
[small caps]	yes	The text is printed as small capital letters.

Tag name	End tag	Description
[strike]	yes	The text is printed strikethrough.
[subscript: size] Do not change the variables, because they are calculated automatically by [lowered: size] Changed or added markup data when dealing with documents originated in [subscript] [lowered] Changed or added markup data when dealing with documents originated in	yes	The text is printed in subscript; the amount of space the text is lowered is given in <i>size</i> .
[superscript: size] Do not change the variables, because they are calculated automatically by [raised:size] Changed or added markup data when dealing with documents originated in [superscript] [raised] Changed or added markup data when dealing with documents originated in	yes	The text is printed in superscript; the amount of space the text is raised is given in <i>size</i> .
[subscript] Changed or added markup data when dealing with documents originated in	yes	The text is printed in subscript.
[superscript] Changed or added markup data when dealing with documents originated in	yes	The text is printed in superscript.
[underline:code] Do not change the variables, because they are calculated automatically by [underline]	yes	The text is printed underlined; the type of underline is given with <i>code</i> .
[vanish][hidden] Changed or added markup data when dealing with documents originated in	yes	The text is not printed because it is hidden; the text is not to be translated.

The following markup has been added to support MS Word 6.0:

[shadow]	yes	The text is printed shadowed.
[bkf: <i>attribute</i>] Do not change the variables, because they are calculated automatically by	no	A bookmark starts at this position in text; where <i>attribute</i> is an index to the bookmark end entry.
[blk: <i>attribute</i>] Do not change the variables, because they are calculated automatically by	no	A bookmark ends at this position in text; where <i>attribute</i> is the value to which the bookmark start points to.
[rev auth: <i>attribute</i>] Do not change the variables, because they are calculated automatically by [rev auth]	yes	Used when revision marking was set to on in Word; is indicator for the author who did the revision.
[DTTM: <i>attribute</i>] Do not change the variables, because they are calculated automatically by [DTTM]	yes	Used when revision marking was set to on in Word; is indicator for date and time when revision took place.
[symb font: <i>name</i> <i>char</i>] Do not change the variables, because they are calculated automatically by	no	Indicates that the character given with <i>char</i> is to be printed in the font given with <i>name</i> .

[csty: <i>name</i>] Do not change the variables, because they are calculated automatically by [csty]	yes	For the run of text enclosed in start and ending tag the style given with <i>name</i> is used.
[kerning: <i>value</i>] Do not change the variables, because they are calculated automatically by [kerning]	yes	For the run of text enclosed in start and ending tag automatic kerning was used; where <i>value</i> is the kerning distance.

Markup data that must not be changed

The following table lists markup data that must not be changed. It is essential that the markup data remains in its original form. Only the position within the segment may change.

When you copy a **Translation Memory** proposal that contains one of these tags and there the source segment does not contain such a tag, you must delete this tag after you have copied the **Translation Memory** proposal.

If your source segment contains such a tag, keep this tag, no matter which type of tags are contained within your **Translation Memory** proposal.

The variables *var1* and *var2* are relevant only for **OpenTM2**.

Tag name	End tag	Description
[auto footnote: <i>ref var1, var2</i>] Changed or added markup data when dealing with documents originated in	no	An autonumbered footnote is inserted; <i>ref</i> is the reference number of the footnote.
[auto endnote: <i>ref var1, var2</i>] Changed or added markup data when dealing with documents originated in	no	An autonumbered end note is inserted; <i>ref</i> is the reference number of the end note.
[efield]	no	The end of a field, even after all nested fields.
[field end: <i>code</i>]	no	The end of the previously inserted field; fields may be nested, the <i>code</i> indicates the type of end field.
[field sep]	no	Field separator.
[field sep: <i>var1, var2</i>]	no	Field separator of embedded objects.
[field: <i>code</i>] Changed or added markup data when dealing with documents originated in	no	Anything that can be inserted with or in Microsoft Word for Windows; <i>code</i> indicates the type of field.
[fld] Changed or added markup data when dealing with documents originated in	yes	A field entry follows.
[footnote: <i>ref var1, var2</i>] Changed or added markup data when dealing with documents originated in	no	A custom footnote is inserted; <i>ref</i> is the reference mark of the footnote.
[endnote: <i>ref var1, var2</i>] Changed or added markup data when dealing with documents originated in	no	A custom referenced end note is inserted; <i>ref</i> is the reference mark of the end note.
[comment ref: <i>var1, var2</i>] Changed or added markup data when dealing with documents originated in	no	An annotation is inserted.
[index entry]	yes	An index entry follows.
[index field] Changed or added markup data when dealing with documents originated in	no	An index entry follows.
[picture: <i>offset</i>]	no	A graphic or picture is inserted in the document; where <i>offset</i> is used by

Tag name	End tag	Description
		OpenTM2 internally.
[object:offset]	no	An object link is inserted in the document; where <i>offset</i> is used by OpenTM2 internally.
[toc entry]	yes	A table of contents entry follows.
[toc field] Changed or added markup data when dealing with documents originated in	no	A table of contents entry follows.
[toa entry]	yes	A table of authority entry follows.
[toa fld]	no	A table of authority entry follows.

Markup data outside translatable segments

The following table lists markup data that is outside of translatable segments. They must not be changed.

The variables *var1* and *var2* are relevant only for OpenTM2.

Tag name	End tag	Description
[Annotation Subdoc]	no	The annotation texts follow.
[column]	no	The start of a new column if the text is written in columns (as in a newspaper).
[comment] Changed or added markup data when dealing with documents originated in	no	An annotation text follows.
[comment text:ref var1, var2] Changed or added markup data when dealing with documents originated in	no	An annotation text follows.
[drawn obj <i>attributes</i>] Changed or added markup data when dealing with documents originated in	no	A drawing object is inserted.
[endnote text:ref var1, var2] Changed or added markup data when dealing with documents originated in	no	The endnote text for the end note reference with <i>ref</i> follows.
[Endnote Subdoc]	no	The end note texts follow.
[footnote text:ref var1, var2]	no	The footnote text for the footnote referenced with <i>ref</i> follows.
[Footnote Subdoc]	no	The footnote texts follow.
[hdr textbox] Changed or added markup data when dealing with documents originated in	no	A header textbox entry.
[header/footer]	no	One header/footer.
[Header/Footer Subdoc]	no	Header/footer texts follow.
[Header Textbox Subdoc]	no	The header textbox texts follow.
[index]	yes	The index follows, it is not translatable. For more information on rebuilding the index, see ...
[lf] Changed or added markup data when dealing with documents originated in	no	Hard line feed.
[next cell:var1, var2]	no	The next cell in a table starts.
[paragraph:var1, var2]	no	Start of a new paragraph.

Tag name	End tag	Description
[row]	no	A new row starts in a table.
[section: <i>var1, var2</i>]	no	A new section starts.
[style: <i>name</i>]	no	Occurs after the paragraph tag and gives the style name for the next paragraph with <i>name</i> .
[subdoc] Changed or added markup data when dealing with documents originated in	no	The current document is a master document and has as subdocument inserted at this text position.
[textbox] Changed or added markup data when dealing with documents originated in	no	A textbox entry.
[Textbox Subdoc]	no	The textbox texts follow.
[toa]	yes	The table of authorities follows, it is not translatable. For more information on rebuilding the table of authorities, see ...
[toc]	yes	The table of contents follows, it is not translatable. For more information on rebuilding the table of contents, see ...

Exchanging data with other OpenTM2 products

Untranslated segments file

During export, you can generate a file with the untranslated segments of a document. This file can be processed by other systems, such as a machine translation system. This file format is similar to an external **Translation Memory**. The following sample contains three segments:

```
<NTMMemoryDb>
<Segment>0000000002
<Control>
000011•0•0000000000000000•English(U.S.)•German(national)•EQFWORD•DEVICE.DOC
</Control>
<Source>Selecting Your Rack-Mounted Devices
</Source>
<Target></Target>
</Segment>
<Segment>0000000009
<Control>
000011•0•0000000000000000•English(U.S.)•German(national)•EQFWORD•DEVICE.DOC
</Control>
<Source>Some items to consider are:
</Source>
<Target></Target>
</Segment>
<Segment>0000000074
<Control>
000011•0•0000000000000000•English(U.S.)•German(national)•EQFWORD•DEVICE.DOC
</Control>
<Source>This publication assists you in selecting a hardware configuration.
</Source>
<Target></Target>
</Segment>
</NTMMemoryDb>
```

Directory structure of OpenTM2

This chapter gives you an overview of the directory structure of a **OpenTM2** installation.

Subdirectory	Content
<i>eqf</i>	All OpenTM2 objects
<i>win</i>	All executable programs
<i>table</i>	All markup tables
<i>prtform</i>	Print-format files used for printing dictionary data
<i>msg</i>	Messages and helps
<i>property</i>	Property files for folders, Translation Memory databases , and dictionaries
<i>export</i>	Subdirectory for exported folders
<i>document</i>	OpenTM2 documentation
<i>mem</i>	Translation Memory databases
<i>dict</i>	Dictionaries
<i>list</i>	Generated lists
<i>samplwp.f00</i>	Folders
<i>\$\$S*****.f00</i>	Temporary folders for source documents from the Initial Translation Memory . This directory is created by the Initial Translation Memory and deleted when the Initial Translation Memory has finished (***** always stands as a variable - these are numbers/characters which are generated freely during ITM process).
<i>\$\$T*****.f00</i>	Temporary folders for target documents from the Initial Translation Memory . This directory is created by the Initial Translation Memory and deleted when the Initial Translation Memory has finished (***** always stands as a variable - these are numbers/characters which are generated freely during ITM process).

The subdirectories with the extension .f00, which contain the folders, have the following structure:

Subdirectory	Content
<i>property</i>	Property files for documents
<i>source</i>	Source documents
<i>ssource</i>	Segmented source files
<i>starget</i>	Segmented target files
<i>target</i>	Target files; only used during the export of documents
<i>eadata</i>	Extended attribute data
<i>mtlog</i>	MT logging information
<i>misc</i>	Intermediate document source files required by some markup tables
<i>RTF</i>	RTF source of MS Word documents

System limitations

This chapter gives an overview of the limits set by the system.

Item	Limit
Folders	unlimited
Translation Memory databases	unlimited
Dictionaries	unlimited
Documents per folder	unlimited
File types per folder	300
Segments in a document	unlimited
Terms in a dictionary	depending on disk size
Segments in a Translation Memory	16,777,215
Markup table size (tags per attribute)	2,000
Segments shown in a OpenTM2 editor	5 to 2.147.483.647
Segments shown in the editor of an Initial Translation Memory	unlimited

Item	Limit
Dictionaries that can be searched at a time	10
Dictionaries that can be used for the analysis at a time	10
	4
Translation Memory databases that can be used for the analysis at a time	10
Maximum number of file pairs in one ITM call	300
Maximum number of file pairs in one ITM creation	900
Setting for right margin	10 to 999
Length of a dictionary headword	254 bytes
Length of a segment in a Translation Memory	2 KB
Maximum size of single-line text that can be pasted	128 bytes
Maximum size of multiline text that can be pasted	512 bytes
Dictionary size	256 MB
Translation Memory size	256 MB
Number of dictionary fields	38
Document size	131 MB

Hints, tips, and technical notices

The following topics are a collection of hints, tips, frequently asked questions, and technical notices that might be of interest in rare cases only. Nevertheless, this information might help to solve or identify one or the other problem.

Where appropriate, other sections in this book refer to these topics. Also, several index entries guide you to these topics; note especially the index entries "hints", "tips", and "technical notices".

About Microsoft^(R) Word documents

Disk space consumption with .doc files

When OpenTM2 processes Microsoft^(R) Word files, it temporarily needs disk space that might be a multiple of the size of the .doc file to be processed. Especially files that contain graphics require more disk space.

The reason is that OpenTM2 uses an internal conversion from .doc to .rtf format to process Microsoft^(R) Word files.

Problem: Word files with embedded picture data cannot be exported

The export of .doc and .rtf documents might fail if they contain embedded objects in a format other than Word's native format. To overcome this problem, convert the embedded objects into graphics before importing.

1. Open a document in Microsoft^(R) Word and locate all pictures, one after the other.
2. Click a picture, and observe the status line.
3. If the status line shows "Double-click to edit Unknown", this picture is not stored in native format.
4. Mark this picture and press CTRL-SHIFT-F9, which converts the embedded object into a graphic.
5. For verification, click the picture again. The "Double-click to edit Unknown" message should not appear.
6. Continue until all pictures are processed this way.

Automatic font conversion for translated RTF documents

OpenTM2 supports the automatic font conversion for translated RTF documents. You can see the conversion when postediting your translation (provided that all the required fonts are installed), or after exporting the translated document.

The specifications for the automatic font substitution are in the language-specific sections of file ..\eqf\table\eqfrtf.chr. If you want to change the specifications, you need to change the parameters following the keywords CHANGEFONT or DEFCHGFONT.

If you want to add specifications for another language, follow the layout of the already existing specifications in this file.

- CHANGEFONT

specifies one or more one-to-one font substitutions.

The general structure of an entry is:CHANGEFONT="source font name]=RTF font spec. for target font" \

(...)

"[last entry]=..." where substitutions are enclosed in double quote pairs, and separated by backslash characters.

The following example:CHANGEFONT="Times New Roman]={f%1\froman\fprq2\fcharset2\fprq2 Arial;" \

"[Courier]={f%1\froman\fprq2\fcharset2\fprq2 Garamond;}"

specifies font substitution from "Times New Roman" to "Arial" and from "Courier" to "Garamond".

The other characters in the *RTF font spec. for target font* specify special characteristics of the substitution.

- DEFCHGFONT

specifies font family substitution.

The general structure of an entry is:DEFCHGFONT="source font family]=RTF font spec. for target font family" \

(...)

"[last entry]=..." where substitutions are enclosed in double quote pairs, and separated by backslash characters.

OpenTM2 supports the following font families as *RTF font spec. for target font family*:font family for automatic font substitution

Table 11. RTF font specification for the target font family

Font family	Characteristics	Examples
\froman	Roman, proportionally spaced serif fonts	Times New Roman, Palatino
\fswiss	Swiss, proportionally spaced sans serif fonts	Arial
\fmodern	Fixed-pitch serif and sans serif fonts	Courier New, Pica
\fscript	Script fonts	Cursive
\fdecor	Decorative fonts	Old English, ITC Zapf Chancery
\ftech	Technical, symbol, and mathematical fonts	Symbol
\fbidi	Arabic, Hebrew, or other bidirectional font	Miriam

Note:

If you use both keywords in a language section, and specify identical source fonts, the one-to-one font substitution takes precedence.

Translating TOC and index sections in Word documents

When you open a Word document in OpenTM2, the table of contents section and the index section of the document are protected and cannot be translated.

To create translated table of contents and index sections, re-create both sections after the translation is completed. Open the document in Microsoft^(R) Word, select where you want to insert the TOC, respectively index, click Index and Tables on the Insert menu, and then click the Table of Contents tab, respectively Index tab. The table of contents and index are then re-created from the (already translated) headings and index entries.

For more details see Microsoft^(R) Word Help.

EQRTF/EQFMSWRD markup table and third-party program limitations

If you use the markup tables EQRTF (for Rich Text Format documents) or EQFMSWRD (for Microsoft^(R) Word documents), note the following limitations:

- The Mac file format is currently not supported.
- Font Embedding is currently not supported.
- For bidirectional languages, note that the Visual C++ help compiler versions 4.01.0950 and 4.03.0002 (part of Visual C++ versions 5.0 and 6.0) do not support the reordering of table columns; therefore table columns are not automatically displayed in reversed order.

bidirectional languageslimitationshelp compiler
limitationshelp compiler

See also [Windows Help/RTF – Table columns not in reverse order](#).

About bidirectional language processing

Windows^(R) Help/RTF – Table columns not in reverse order

Although the RTF specification encloses special tags for bidirectional language processing of tables, the Microsoft^(R) Help compiler does not support column reordering for bidirectional languages. Note the following excerpt from a Microsoft^(R) technical article: "... Right-to-left tables, a standard feature of word processors such as Arabic Microsoft^(R) Word, are not a supported feature of Help Compiler Workshop. Help Compiler Workshop will compile a right-to-left table as if it is a left-to-right table.".

See also [EQFRTF/EQFMSWRD markup table and third-party program limitations](#).

As a workaround, you can edit the tables manually before importing the documents into OpenTM2. You can use the Word macro function to automate this task.

Problem determination

This chapter helps you solve problems that might arise when working with **OpenTM2**.

...

Accessibility

Accessibility features help users with physical disabilities, such as restricted mobility or limited vision, to use software products successfully. The following list describes the major accessibility features of OpenTM2:

- You can use keyboard shortcuts instead of the mouse to perform the most common tasks.
- You can modify the properties to use colors for the different text types of OpenTM2.
- You can modify the properties to display different font styles or font sizes.
- You can assign keys (new, change and/or delete) to almost all available functions OpenTM2 provides to the translator.

The following sections explain how to use these accessibility features.

This table describes the File menu in the main view:

Table 12. File Menu 1

Menu bar choices	Actions	Keyboard shortcuts
File => Open	Opening the 'Document List' window from selected folder (works only in the deselected 'Use Explorer-like tree view')	Alt+O
File => New...	Creating <ul style="list-style-type: none">• a new subfolder (when a file is selected in the 'Folder List' window).• new folder (when a folder is selected in the 'Folder List' window).• a translation memory (when 'Translation Memory' window is selected).• a dictionary (when 'Dictionary List' window is selected).• a new markup table (when 'Markup Table List' window is selected).• a new exclusion (when 'Exclusion Lists' window is selected).	Alt+N

Menu bar choices	Actions	Keyboard shortcuts
File => Properties	<p>Opening</p> <ul style="list-style-type: none"> • the 'Folder Properties' window (when a folder is selected in the 'Folder List' window). • the 'Document Properties' window (when a file is selected in the 'Folder List' window). • the 'Translation Memory Properties' window (when the 'Translation Memory' window is selected). • the 'Dictionary Properties' window (when the 'Dictionary List' window is selected). • the 'Markup Table Properties' window (when the 'Markup Table List' window is selected). 	Alt+r
File => Properties Summary	<p>Opening</p> <ul style="list-style-type: none"> • the 'Folder Properties Summary' window (when a folder is selected in the 'Folder List' window). • the 'Document Properties Summary' window (when a file is selected in the 'Folder List' window). • the 'Translation Memory Properties Summary' window (when the 'Translation Memory' window is selected). • the Dictionary Properties Summary window (when the 'Dictionary List' window is selected). 	Alt+u
File => Delete	Deleting the selection (e.g. TM folder, file, memory or dictionary).	Alt+d
File => Rename Object	Renaming the selected object (e.g. TM folder, memory or dictionary).	Alt+b
File => Cut	Cutting to the clipboard.	—
File => Copy	Copying to the clipboard.	—
File => Paste	Pasting to the clipboard.	—
File => Merge...	Merging selected translation memory from the 'Translation Memory List' window with another selectable memory.	Alt+m
File => Organize	Organizing selected translation memory or dictionary.	Alt+g
File => Analyze...	Analyzing the selected OpenTM2 folder.	Alt+a
File => Print...	Printing the file content	Alt+p

Menu bar choices	Actions	Keyboard shortcuts
	opened in the translation view window.	
File => Print List...	Printing the file list of an OpenTM2 folder.	Alt+t
File => System Preference	Opening the 'System Preferences' window.	Alt+y
File => Export	Exporting selected file (e.g. OpenTM2 folder, file, memory or dictionary).	Alt+E
File => Import	Importing file (e.g. OpenTM2 folder, file, memory or dictionary).	Alt+I
File => Global Find and Replace	Opening the 'Global Find and Replace' window to find and change a term in all files of a selected folder.	Alt+F
File => Select All	Selecting all files in a folder.	Alt+S
File => Deselect All	Deselecting all files in a folder.	Alt+I

This table describes the File menu in the translation view:

Table 13. File Menu 2

Menu bar choices	Actions	Keyboard shortcuts
File => Translation Environment	Selecting one of the file(s) which was/were opened in the translation view.	Alt+T
File => Open...	Opening another document for checking purposes while translating.	Alt+O
File => Save	Saving the current file in the translation view.	Alt+S
File => Print	Printing the current file in the translation view.	Alt+P
File => Quit	Quitting the current file in the translation view.	Alt+Q
File => End-Save	Saving and quitting the current file in the translation view.	Alt+E
File => Translation Window	Bringing the 'Translation' window to front.	Alt+1
File => Translation Memory Window	Bringing the 'Translation Memory' window to front.	Alt+2
File => Dictionary Window	Bringing the 'Dictionary' window to front.	Alt+3
File => Original Window	Bringing the 'Original' window to front.	Alt+4
File => Source of Proposal Window	Bringing the 'Source of Proposal' window to front.	Alt+5

Colors, fonts and key assignment

In all OpenTM2 versions, you can change the font color, background color and font styles of each segment (translated, untranslated and coded text parts) and proposals. These changes can be performed by the following windows:

- Translations window
- Translation Memory window
- Source of Proposal(s) window
- Original window
- Dictionary window

By default, the segments are presented in colors and fonts that can be easily differentiated by color-blind users. For example, most of the color-blind people have difficulties in differentiating

between green and red colors. Therefore, these colors were not used as default values.

You can modify the default settings by clicking **Options => Color...** from the menu bar (it is necessary to activate the translation view).

Figure 166. Set Colors Window

[PIC]Figure shows Set Colors Window

You can modify the default settings by clicking **Options => Fonts...** from the menu bar (it is necessary to activate the translation view).

Figure 167. Set Fonts Window

[PIC]Figure shows Set Fonts Window

You can modify the default settings by clicking **Options => Keys...** from the menu bar (it is necessary to activate the translation view).

Figure 168. Assign Keys Window

[PIC]Figure shows Assign Keys Window

Set colors



Set fonts



Assign keys



Keyboard shortcuts

OpenTM2 provides you can use keyboard shortcuts for the most frequently used tasks. These keyboard shortcuts are described in the following sections.

Table 14. Opening menus

Task	Keyboard shortcut
Moving the focus to the menu bar	F10

Table 15. Navigating in windows

Task	Keyboard shortcut
Moving the cursor one selection down/up	Down/Up arrow
Moving the cursor one screen view down/up	Page Down/Up
Moving the cursor to first entry in the window	Home
Moving the cursor to last entry in the window	End

Table 16. Default keyboard shortcuts in Translation window

Task	Keyboard shortcut
Save	F2
Quit	F3
End-Save	F4
Translation Memory Window	F8
Copy	Ctrl+c
Paste	Ctrl+v
Cut	Ctrl+x
Undo	Alt+Backspace
Start of Segment	Alt+Home
End of Segment	Alt+End
Start of Line	Home
End of Line	End
Start of Segment	Alt+Home
End of Segment	Alt+End

Task	Keyboard shortcut
Translate Segment	Ctrl+Newline
Next Untranslated Segment	Ctrl+n

Table 17. Help

Task	Keyboard shortcut
Opening the online help	F1

Table 18. Special accessibility keyboard shortcuts

Task	Keyboard shortcut
Toggles StickyKeys on and off	Tap <SHIFT> 5 times
Toggles FilterKeys on and off	Hold the <Right SHIFT> key for 8 seconds
Toggles ToggleKeys on and off	Hold the <NUM LOCK> key for 5 seconds
Toggles MouseKeys on and off	<Left ALT>+<Left SHIFT>+<NUM LOCK>
Toggles High Contrast on and off	<Left ALT>+<Left SHIFT>+<PRINT SCREEN>

Opening menus

...

Navigating in windows (Folder List window, Document List window, Translation Memory List window or Dictionary List window)

...

Default keyboard shortcuts in Translation window

...

Help

...

Special accessibility keyboard shortcuts

...

Menu bar choices

This chapter summarizes the menu bar choices of OpenTM2. The first five menu items

- File (Alt+F)
- View (Alt+V)
- Utilities (Alt+U)
- Window (Alt+W)
- Help (Alt+H)

are visible for all windows.

The following seven menu items are only visible when the translation window is open:

- File (Alt+F, this File menu contains different choices than the File menu in the main view)
- Edit (Alt+E)
- Options (Alt+O)
- Cursor (Alt+r)
- Translate (Alt+T)
- Spellcheck (Alt+p)
- Style (Alt+l)

As a prerequisite for the Actions described below, the view of OpenTM2 must be set to 'Use Explorer-like tree view'.

File Menu 1



File Menu 2



View Menu

Table 19. View Menu

Menu bar choices	Actions	Keyboard shortcuts
View => Name	Making only the 'Name' Column visible.	Alt+N
View => Details	Making all selected details visible.	Alt+D
View => Change Details...	Changing the details view.	Alt+C
View => Sort...	Opening the 'Sort List Item' window.	Alt+o
View => Filter...	Opening the 'Filter List Items' dialog window.	Alt+F
View => Show All	Showing all files, canceling the filter function.	Alt+A
View => Shrink Path	Shrinking the path information for all files.	—
View => Hide Path	Hiding the path information for all files.	—
View => Show Path	Showing the path information for all files.	—
View => Toolbar	Selecting one of the following settings: <ul style="list-style-type: none">• Workbench• Translation Environment• Status bar	Alt+T

Utilities Menu

Table 20. Utilities Menu

Menu bar choices	Actions	Keyboard shortcuts
Utilities => Count Words...	Opening the 'Count Words' window (TranslationManger folder or file based).	Alt+W
Utilities => Create Counting Report...	Opening the 'Create Counting Report' window (TranslationManger folder or file based).	Alt+R
Utilities => Configure Drives...	Configuring drives.	Alt+D
Utilities => Connect Shared Resources...	Connecting shared resources.	Alt+o
Utilities => Build Archive Translation Memory...	Building an archive translation memory from the selected folder.	Alt+B
Utilities => Display Markup Table List	Displaying 'Markup Table List' window.	Alt+M
Utilities => Display Language List	Displaying 'Language List' window.	Alt+L
Utilities => Display Terminology List	Displaying 'Terminology List' window.	Alt+T
Utilities => Display Machine Translation Job List	Displaying 'Machine Translation Job List' window.	—

Window Menu

Table 21. Window Menu

Menu bar choices	Actions	Keyboard shortcuts
Window => Minimize All	Minimizing all open windows.	Alt+M
Window => Restore All	Restoring all active windows.	Alt+R
Window => Tile Window	Tiling all windows in the OpenTM2 Workbench.	Alt+T
Window => Cascade Window	Cascading all windows in the OpenTM2 Workbench.	Alt+C
Window => 1) Folder List	Bringing the 'Folder List' window to front.	Alt+1
Window => 2) Translation Memory List [Some]	Bringing the 'Translation Memory List [Some]' window to front.	Alt+2
Window => 3) Dictionary List [Some]	Bringing the 'Dictionary List [Some]' window to front.	Alt+3

Help Menu

Table 22. Help Menu

Menu bar choices	Actions	Keyboard shortcuts
Help => Help for Help...	Opening the 'Help Topics' window.	Alt+H
Help => Help Index...	Opening the 'OpenTM2 Help' window.	Alt+I
Help => Homepage	Opening the TM intranet homepage via standard web browser.	Alt+p
Help => Documentation (PDF)	Opening the current TM via Acrobat Reader (PDF file format).	Alt+D
Help => Documentation (HTML)	Opening the current TM via standard web browser (HTML file format).	Alt+o
Help => Product Information...	Opening the 'Product Information' window.	Alt+P

Edit Menu

Table 23. Edit Menu

Menu bar choices	Actions	Keyboard shortcuts
Edit => Find and Replace...	Opening the 'Find and Replace' window for the 'Translation' window.	Alt+F
Edit => Cut	Cutting to the clipboard.	Alt+t
Edit => Copy	Copying to the clipboard.	Alt+C
Edit => Paste	Pasting to the clipboard.	Alt+P
Edit => Clear	Clearing the selection.	Alt+l
Edit => Undo	Undoing previous action.	Alt+U
Edit => Copy Proposal Block	Copying selected proposal block.	Alt+o
Edit => Unmark Block	Removing marks from text section.	Alt+m
Edit => Split Line	Splitting the line at the position of cursor.	Alt+S
Edit => Join Line	Joining the line at the position of cursor.	Alt+J
Edit => Line Wrap	Setting the wrapping at margin column on or off.	Alt+W
Edit => Reflow Segment	Reflowing the current segment is refloew	Alt+R

Menu bar choices	Actions	Keyboard shortcuts
	(depending on definition of right margin). This function is not active if right margin is set to AUTO.	

Options Menu

Table 24. Options Menu

Menu bar choices	Actions	Keyboard shortcuts
Options => Color...	Opening the 'Set Colors' window.	Alt+C
Options => Fonts...	Opening the 'Set Fonts' window.	Alt+F
Options => Keys...	Opening the 'Assign Keys' window.	Alt+K
Options => Sentence Lookup...	Opening the 'Sentence Lookup' window.	Alt+S
Options => Commands...	Opening the 'Run Command' window.	Alt+o
Options => Profile Settings...	Opening the 'Customize Translation Functions' window.	Alt+P
Options => Tailor	Selecting one of the following settings: • Ruler • Horizontal • Vertical • Titlebar • Status bar	—

Cursor Menu

Table 25. Cursor Menu

Menu bar choices	Actions	Keyboard shortcuts
Cursor => Next Updated	Jumping to next updated segment.	—
Cursor => Find block	Moving cursor to marked text section.	Alt+F
Cursor => Special Go To...	Opening 'Special Go To' window.	Alt+p
Cursor => Go to Line...	Opening the 'Go to Line' window.	Alt+G
Cursor => Go to Segment	Opening the 'Go to Segment' window.	
Cursor => Query Line	Querying line information of selected line.	Alt+Q
Cursor => Top	Placing the cursor at the start of the file.	Alt+T
Cursor => Bottom	Placing the cursor at the end of the file.	Alt+B
Cursor => Start of Line	Placing the cursor at the start of the line.	Alt+S
Cursor => End of Line	Placing the cursor at the end of the line.	Alt+E
Cursor => Start of Segment	Placing the cursor at the start of the segment.	Alt+a
Cursor => End of Segment	Placing the cursor at the end of the segment.	Alt+n

Translation Menu

Table 26. Translation Menu

Menu bar choices	Actions	Keyboard shortcuts
Translation => Translate Segment	Confirming a translation and activating next segment.	Alt+T
Translation => Next Untranslated Segment	Jumping to next untranslated segment.	Alt+N
Translation => Untranslate Segment	Deleting the translated segment from translation memory.	Alt+U
Translation => Look Up a Term...	Opening the 'Look Up a Term' window.	Alt+L
Translation => Edit a Term...	Opening the 'Edit Entry in Dictionary' window.	Alt+E
Translation => Add an Abbreviation	Adding selected word to the abbreviation list.	—
Translation => Edit Abbreviations...	Editing the abbreviation list.	—
Translation => Go To Active Segment	Going to active segment.	Alt+G
Translation => Join Segments	Joining segments.	Alt+J
Translation => Split Joined Segments	Splitting joined segments.	Alt+S
Translation => Set Bookmark	Setting a bookmark into the translation window (segmentation based).	Alt+B
Translation => Go To Bookmark	Going to a created bookmark.	Alt+k
Translation => Clear Bookmark	Clearing created bookmarks.	Alt+C
Translation => Postediting	Reviewing the translation and checking the spelling.	Alt+P
Translation => Automatic Substitution	Copying automatically existing translated segments into the translation document as long as exact matches for the source segments are found.	Alt+A
Translation => Show Translation	Showing translated file in a preview window.	—

Spellcheck Menu

Table 27. Spellcheck Menu

Menu bar choices	Actions	Keyboard shortcuts
Spellcheck => Segment...	Spellchecking the selected segment.	Alt+S
Spellcheck => File...	Spellchecking the selected file.	Alt+F
Spellcheck => Edit Addendum Terms...	Opening the 'Edit Addendum Terms for:' window for the target language.	—
Spellcheck => Auto Spellcheck	Turning the auto spellcheck on or off.	Alt+A
Spellcheck => Next Misspelled	Jumping to next misspelled word.	Alt+N

Style Menu

Only one item can be selected at a time

Table 28. Style Menu

Menu bar choices	Actions	Keyboard shortcuts
Style => - Protect - Unprotect - Hide	Changing the style in the 'Translation' window.	- Alt+P - Alt+U - Alt+H - Alt+S

Menu bar choices	Actions	Keyboard shortcuts
- Shrink - Compact - Compact+1		- Alt+C - Alt+o

Trademarks

The following terms are trademarks of the IBM Corporation in the United States, other countries, or both:

- BookMaster
- CUA
- IBM
- OS/2
- Presentation Manager
- VTAM

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Lotus Notes is a registered trademark of Lotus Development Corporation.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Glossary of terms and abbreviations

This glossary defines all new terms and abbreviations used in this manual. It does not include all terms previously established for operating systems. If you do not find the term you are looking for, refer to the index or to the *Dictionary of Computing*, New York: McGraw-Hill, 1994.

- **addendum**

The extension of a *language-support file* that contains individually added spellings of terms. For example, terms which have been indicated as misspelled by the spellchecker although spelled correctly.

- **aligning**

The process of combining source segments with their corresponding target segments in an Initial Translation Memory (ITM).

- **analysis**

A process for dividing text into *segments*. It checks the text against specific *exclusion lists* and *dictionaries*, and produces, on your request, a *new terms list* and a *found terms list*.

- **ANSI**

American National Standards Institute.

- **API**

Application programming interface.

- **application programming interface (API)**

A software interface that enables applications to communicate with each other. An API is the set of programming language constructs or statements that can be coded in an application program to obtain the specific functions and services provided by an underlying operating system or service program.

In VTAM^(R), the language structure used in control blocks so that application programs can reference them and be identified to VTAM^(R).

- **automatic lookup**

During translation, **OpenTM2** performs an automatic lookup in the referenced **Translation Memory** and in the referenced *dictionaries*. For each segment, matching segment translations from the **Translation Memory** are displayed as *translation proposals* in the "**Translation Memory**" window, translations of its terms are displayed in the "Dictionary" window.

- **automatic substitution**

An option in the Translate menu. It lets you start the automatic substitution process, which translates those *segments* that have been previously translated by you or another translator and are stored in the *Translation Memory*. It is particularly useful for translating updated text. However, you still must translate new text manually.

- **company code**

Abbreviation for a particular area of usage a translation applies to. For example, certain terms are used differently depending on the companies or clients you do translations for.

- **controlled folder handling**

Is a concept that is only available to project coordinators. It allows them to specify, and change at any time, all properties and details for a folder, including the translators for the documents to be imported into this folder. It also allows them to ship the folder once all translations are finished.

- **details**

See *view details*.

- **dictionary**

A database that contains terms, their translation, and other related information.

- **dictionary entry**

All data relating to a *headword* in a *dictionary*

- **dictionary filter**

A method to select specific entries from a *dictionary* or only parts of these entries. The filter conditions that must be met if an entry is to pass the filter can be individually defined when printing or searching a dictionary.

- **dictionary print format**

Specifies the layout of a printed *dictionary*. **OpenTM2** provides standard formats described in *format files* that can be tailored individually. The format files are on the same disk where **OpenTM2** is installed under the subdirectory `eqf\prtform`.

- **DLL**

Dynamic-link library.

- **document file**

A generic term used to describe all types of files containing information that is to be translated. Document files can be analyzed and opened for translation in the *Translation Environment*. The source of the document file you translate is called the *original document*. The document file that you edit during translation is referred to as the *translation document*.

- **document type**

Depending on the different types of *markup* used to describe the layout of document, **OpenTM2** differentiates between different document types.

- **dynamic-link library (DLL)**

A file containing executable code and data bound to a program at load time or runtime, rather than during linking. The code and data in a dynamic-link library can be shared by several applications simultaneously.

- **entry fields**

The various fields and styles of an entry in a *dictionary*, such as meaning, usage, context, abbreviation, idioms, and grammatical information. For example, the entry field *Abbr.* would contain the abbreviation of a *headword*. The combination of all entry fields of a specific headword makes up the headword's entry in the dictionary.

- **entry level**

The information that applies to all the *templates* of an entry. For example, the term itself, the author, and the date the entry was created.

- **entry section**

Section in a *dictionary*. Contains all *dictionary entries* appearing one after another.

- **exact match**

Each *segment* in the *translation document* is compared with the selected *Translation Memory*. If an identical segment is found, an *exact match* has occurred and the corresponding *translation proposal* is shown in the "Translation Memory" window. It originates from a previous translation.

- **exact match (1)**

An *exact match* for which the following condition applies: The exact match occurs only once in the attached **Translation Memory databases**.

- **exact match (>=2)**

An *exact match* for which the following condition applies: The exact match occurs at least twice in the attached **Translation Memory databases**.

- **exact-exact match**

An *exact match* for which the following condition applies: The number of the active segment in the source document is identical (give or take 2) with that of the corresponding segment in the

Translation Memory. In addition, the name of the document (document name = file name plus relative path (if available)) being translated is identical with that of the document stored in the Translation Memory.

- **exact context match**

An *exact match* for which the following condition applies: The number of the active segment in the source document is not identical with that of the corresponding segment in the Translation Memory. However, the name of the document being translated is identical with that of the document stored in the Translation Memory.

- **exclusion list**

A list containing common words such as articles, prepositions, proper nouns, and terms that occur frequently. These words are ignored when creating *new terms lists* and *found terms lists* during *analysis*, and are not shown in the "Dictionary" window during translation. Exclusion lists can be edited.

- **export**

To copy *folders*, documents, *dictionaries*, and *Translation Memory databases* to the DOS file system to make them available to another user.

- **folder**

Contains documents belonging to one project and references to the **Translation Memory databases** and *dictionaries* you want to use during translation.

- **format file**

A file that contains the specification of a *dictionary print format*. It can be created and changed with a text editor.

- **found terms list**

A list of all terms in the documents being analyzed that were found in the selected *dictionaries*. The list is used to update dictionaries and *exclusion lists*. Found terms lists can be edited, that is, terms can be deleted, moved to a dictionary, or to an *exclusion list*. A found terms list can be used to fill a separate dictionary related to a document.

- **fuzzy match**

Each *segment* in the *translation document* is compared with the selected *Translation Memory*. If an almost identical segment is found, a fuzzy match has occurred and the corresponding *translation proposal* is shown in the "Translation Memory" window with a preceding [f]. It originates from a previous translation.

- **fuzzy replacement match**

A *replacement match* where a couple of words are not identical. It is displayed in the "Translation" window with a preceding [rf]. Example: Document text: This is what happened in 1998.

TM proposal: This happens in 1999. In this example, the date in the TM proposal (1999) is automatically changed to the date in the document text (1998). However, happened is not replaced with happens.

- **header section**

Section in a *dictionary*. Contains general dictionary information such as source language, target language, and creation date of the dictionary.

- **headword**

Word or term placed at the beginning of an entry in a *dictionary*.

- **history log file**

A file storing, in compressed form, records that contain the information collected during events, such as exporting or deleting a folder, and the result of this collection. There is one history log file per folder, which is stored as `HISTLOG.DAT` in the `PROPERTY` directory of the folder. New records are added at the end of the history log file.

- **homonym**

Words that are spelled and pronounced alike but different in meaning. For example, the noun conduct and the verb conduct are homonyms.

- **homonym level**

Part of a *dictionary entry*. Contains grammatical and syntactic information, such as part of speech, hyphenation, and abbreviation information.

- **HTML**

Hypertext Markup Language.

- **Hypertext Markup Language (HTML)**

A subset of the Standard Generalized Markup Language (SGML) allowing the presentation of

electronically stored information within the World Wide Web (Internet).

- **icon**

A small graphical symbol. Icons can represent windows that you want to work with (such as Folder list, Document list, Dictionary list, Translation Memory list, Terminology lists) or tasks that you want to perform.

- **import**

To copy *folders*, documents, *dictionaries*, and *Translation Memory databases* from the DOS file system to make them available to **OpenTM2**.

- **Initial Translation Memory (ITM)**

A *Translation Memory* created from existing translations and their corresponding originals. Proposals originating from an ITM are shown in the "Translation Memory" window with a preceding [m] like *machine-generated matches*.

- **irregular match**

One of the following:

- A 1:2 match, where one source segment has been connected to two target segments
- A 2:1 match, where two source segments have been connected to one target segment
- A 2:2 match, where two source segments have been connected to two target segments
- An unaligned sentence (the default color is red)
- A sentence that is ignored (the default color is grey)

- **ITM**

Initial Translation Memory.

- **JavaScript**

A scripting language that resembles Java[™] and was developed by Netscape for use with the Netscape browser.

- **language support files**

Source languages supplied with **OpenTM2**. Language support files are required when looking up *dictionary entries* during *analysis* of document files and during *spellcheck*.

- **lookup**

See *automatic lookup* and *search*.

- **machine-generated match**

Originates from an *Initial Translation Memory* and is displayed in the "Translation Memory" window with a preceding [m]. Can be used in the same way as a *fuzzy match*.

- **mappable section**

Section in a *dictionary*. Determines the structure of *dictionary entries*. Contains the total of all allowed entry fields in a dictionary.

- **markup**

Information added to a document, for example, formatting tags, to enable a system to process it. It describes the document characteristics or specifies the actual processing to be performed.

- **markup language**

The language specific to a word processor that describes a document layout.

- **markup table**

Contains all tags and attributes of a particular *markup language*. Is used in **OpenTM2** during *analysis* and translation.

- **match**

The fact that a source *segment* in a **Translation Memory** and a source segment in a document to be translated at least resemble each other (*fuzzy match* or *replacement match*). If they are completely identical, it is an *exact match* if the translation was done by a translator, or a *machine-generated match* if the translation is generated by a program.

- **merge**

Combining information of either two *dictionaries* or two *Translation Memory databases*. When merging dictionaries, **OpenTM2** preserves the structure of the destination dictionary.

- **model dictionary**

An already existing *dictionary* whose structure can be taken as a sample when creating a new dictionary.

- **model folder**

An already existing *folder* whose *properties* can be taken as a sample when creating a new folder.

- **new terms list**
A list of all the terms found in the documents being analyzed but not found in the selected *dictionaries* during *analysis*. New terms lists can be used to update dictionaries and *exclusion lists*. New terms lists can be edited, that is, terms can be deleted, moved to a dictionary, or to an exclusion list.
- **organize**
Internal restructuring of frequently changed *dictionaries* and **Translation Memory databases** to shorten search times.
- **original document**
The source of the document that you translate. You cannot edit this document but you can display it and use it for comparison or checking purposes.
- **postediting**
Editing an already translated document. Any changes cause an automatic update of the already translated *segments* in the *Translation Memory*.
- **properties**
A summary of the different characteristics of a *folder* or a document, such as a description, the *markup language* used in documents, and references to **Translation Memory databases** and *dictionaries*.
- **replacement match**
An *exact match* where only a number or date differs. It is displayed in the "Translation" window with a preceding [r]. Example: Document text: This happened in 1998.

TM proposal: This happened in 1999. In this example, the date in the TM proposal (1999) is automatically changed to the date in the document text (1998).
- **reversing**
Turning source segments contained in a Translation Memory into target segments and vice versa.
- **revision marks**
Characters at the beginning and end of a *segment* that can be individually defined and indicate that the enclosed segment has been translated from scratch, or by copying a *translation proposal* and changing it, or by copying a proposal without changing it.
- **search**
In the "Look up a Term" window, you can search for terms in a dictionary using predefined search criteria and user-definable *dictionary filters*. See also *automatic lookup*.
- **segment**
A translation unit produced during *analysis*. It is usually a sentence, part of a sentence, an element of a list, or a citation.
- **sense level**
Part of a *dictionary entry*. Contains semantic variations of a *headword* such as varying areas of meaning and usage.
- **SGML**
Standard Generalized Markup Language.
- **shared translation material**
A dictionary or Translation Memory file located on a shared disk. It can be concurrently accessed by all **OpenTM2** users who are connected to the same LAN.
- **source document**
See *original document*.
- **spellcheck**
A proofreading aid to identify unrecognized or misspelled words in *translation documents*. Lists possible corrections for misspelled words.
- **Standard Generalized Markup Language (SGML)**
A set of rules that allows the format specification of a *markup language* independent of any individual processing system. The external file formats created during export are based on SGML.
- **stem**
The part of an inflected word that remains unchanged except by phonetic changes or variations throughout an inflection.

- **subject code**
Abbreviation for a particular subject area a translation applies to.
- **tag**
Statement used to determine the format of a *document file*. Is contained in a *markup table*.
- **target document**
See *translation document*.
- **target level**
Contains all information applying to one translation variant of a *headword*, such as definition and usage.
- **template**
Dictionary entry information on all levels (*entry, homonym, sense, and target*) relating to one specific translation of a *headword*.
- **terminology list**
A generic term for the following types of lists: *exclusion lists, found terms lists, and new terms lists*.
- **translation document**
The document that you translate.
- **Translation Environment**
Environment where the actual translation is performed. It consists of a window where you can edit the document file, a window with proposals from the associated *Translation Memory*, and a window with translations for terms in the document. All *translation proposals* can be copied into the *translation document*.
- **Translation Memory**
A database that contains previously translated *segments* added during translation and *analysis*.
- **Translation Memory databases**
More than one **Translation Memory**.
- **translation proposal**
The translation of a *segment* found in a **Translation Memory** during translation, where the source segment is identical (*exact match*) or almost identical (*fuzzy match*) to the current segment.
- **user exit**
A point in an OpenTM2-supplied program at which a user exit routine may be given control.
A programming service provided by an software product that may be requested during the execution of an application program for the service of transferring control back to the application program upon the later occurrence of a user-specified event.
- **view details**
Contents of the list windows displayed in the main window. You can define how detailed the contents of these lists is to be displayed. The default is to display only the names of the individual list items.
- **word count**
Utility to count words (words to be translated, words already translated, *markup tags*) in *original documents* or *translation documents*.
- **workbench**
The **OpenTM2** main window.

User-Defined Index

<i>Translation Environment</i>	<i>bidi</i> irectional languages	19
bidi iirectional languages.....	19	
<i>bidi</i> irectional languages.....	19	
<i>Arabic languages</i>	see <i>bidi</i> irectional languages	19
<i>see bidi</i> irectional languages.....	19	
<i>Hebrew languages</i>	see <i>bidi</i> irectional languages	19
<i>see bidi</i> irectional languages.....	19	
<i>customizing</i>	for <i>bidi</i> irectional languages	20
<i>for bidi</i> irectional languages.....	20	
<i>visual display option</i>	20	
<i>text alignment</i>	<i>Arabic language</i>	22
<i>Arabic language</i>	22	
<i>reading order</i>	<i>Arabic language</i>	22
<i>Arabic language</i>	22	
<i>reading order</i>	22	
<i>text alignment</i>	22	
<i>Arabic language</i>	<i>shaping</i>	22
<i>shaping</i>	22	
<i>shaping of Arabic characters</i>	22	
<i>of Arabic characters</i>	22	
<i>Arabic language</i>	<i>ligating</i>	22
<i>ligating</i>	22	
<i>ligating of Arabic characters</i>	22	
<i>of Arabic characters</i>	22	
<i>text alignment</i>	<i>Hebrew language</i>	23
<i>Hebrew language</i>	23	
<i>reading order</i>	<i>Hebrew language</i>	23
<i>Hebrew language</i>	23	
<i>Hebrew language</i>	<i>reading order text alignment</i>	23
<i>reading order</i>	23	
<i>text alignment</i>	23	
<i>copying</i>	<i>Translation Memory proposals</i>	26
<i>Translation Memory proposals</i>	26	
<i>spellchecking</i>	<i>Thai documents</i>	26
<i>Thai documents</i>	26	
<i>Thai language</i>	28	
<i>Thai language</i>	<i>abbreviation list</i>	28
<i>abbreviation list</i>	28	
<i>RTFEdit editor</i>	31	
<i>cursor moving</i>	33	
<i>moving</i>	33	
<i>copying</i>	<i>Translation Memory proposals</i>	35
<i>Translation Memory proposals</i>	35	
<i>Translation Memory copying proposals</i>	35	
<i>copying proposals</i>	35	
<i>editor functions</i>	<i>deleting characters</i>	36
<i>deleting characters</i>	36	
<i>deleting characters</i>	36	
<i>characters</i>	36	
<i>editor functions</i>	<i>deleting lines</i>	36
<i>deleting lines</i>	36	
<i>deleting lines</i>	36	
<i>lines</i>	36	
<i>editor functions</i>	<i>deleting text until next markup tag</i>	36
<i>deleting text until next markup tag</i>	36	
<i>deleting text until next markup tag</i>	36	
<i>text until next markup tag</i>	36	
<i>editor functions</i>	<i>deleting words</i>	36
<i>deleting words</i>	36	
<i>deleting words</i>	36	
<i>words</i>	36	
<i>editor functions</i>	<i>find and replace terms</i>	36
<i>find and replace terms</i>	36	
<i>finding and replacing terms</i>	36	
<i>lines</i>	<i>inserting</i>	36
<i>inserting</i>	36	
<i>inserting a line</i>	36	
<i>editor functions</i>	<i>inserting a line</i>	36
<i>inserting a line</i>	36	
<i>lines</i>	<i>joining</i>	36
<i>joining</i>	36	
<i>joining lines</i>	36	
<i>editor functions</i>	<i>joining lines</i>	36
<i>joining lines</i>	36	

<i>linesplitting</i>	36
<i>splitting</i>	36
<i>splitting a line</i>	36
<i>editor functions</i>	36
<i>splitting lines</i>	36
<i>linetruncating</i>	36
<i>truncating</i>	36
<i>truncating lines</i>	36
<i>editor functions</i>	36
<i>truncating lines</i>	36
<i>undoing an action</i>	36
<i>editor functions</i>	36
<i>undoing an action</i>	36
<i>lineswrapping</i>	36
<i>wrapping</i>	36
<i>wrapping lines</i>	36
<i>editor functions</i>	36
<i>wrapping lines</i>	36
<i>abbreviations</i>	36
<i>adding new ones</i>	36
<i>automatic substitution</i>	36
<i>dictionary</i>	36
<i>copying proposals</i>	36
<i>copying proposals</i>	36
<i>copying dictionary proposals</i>	36
<i>dictionary proposals</i>	36
<i>copying Translation Memory proposals</i>	37
<i>Translation Memory proposals</i>	37
<i>Translation Memory</i>	37
<i>copying proposals</i>	37
<i>deleting Translation Memory proposals</i>	37
<i>Translation Memory proposals</i>	37
<i>Translation Memory</i>	37
<i>deleting proposals</i>	37
<i>abbreviations</i>	37
<i>updating</i>	37
<i>spellchecker addendum, updating</i>	37
<i>looking up sentences in a Translation Memory</i>	37
<i>sentences in a Translation Memory</i>	37
<i>Translation Memory</i>	37
<i>looking up sentences</i>	37
<i>looking up sentences</i>	37
<i>looking up terms in a dictionary</i>	37
<i>terms in a dictionary</i>	37
<i>editing dictionary entries</i>	37
<i>dictionary entries</i>	37
<i>dictionary</i>	37
<i>looking up terms</i>	37
<i>looking up terms</i>	37
<i>editing entries</i>	37
<i>posting</i>	37
<i>translations</i>	37
<i>revising</i>	37
<i>translations</i>	37
<i>spellchecking</i>	37
<i>spellchecking</i>	37
<i>toggling first character of dictionary</i>	37
<i>editor functions</i>	37
<i>toggling first character of dictionary proposal</i>	37
<i>toggling first character of dictionary proposal</i>	37
<i>dictionary</i>	37
<i>toggling first character of proposal</i>	37
<i>translations</i>	38
<i>referring to other documents while translating</i>	38
<i>referring to other documents while translating</i>	38
<i>document</i>	38
<i>opening more documents while translating</i>	38
<i>opening more documents while translating</i>	38
<i>document</i>	38
<i>displaying original while translating</i>	38
<i>displaying original while translating</i>	38
<i>original document, viewing</i>	38
<i>Translation Memory matches</i>	38
<i>displaying corresponding sources</i>	38
<i>displaying corresponding sources</i>	38
<i>matches</i>	38
<i>displaying corresponding sources</i>	38
<i>displaying corresponding sources</i>	38
<i>Translation Environment</i>	38
<i>displaying tables of contents</i>	38
<i>displaying tables of contents</i>	38
<i>document</i>	38
<i>displaying tables of contents</i>	38
<i>displaying tables of contents</i>	38
<i>viewing</i>	38
<i>tables of contents</i>	38
<i>displaying tables of contents</i>	38
<i>tables of contents</i>	38
<i>table of contents</i>	38
<i>displaying</i>	38

displaying.....	38
Translation Environment	38
displaying translations.....	38
document	38
displaying translations.....	38
viewing	38
translations.....	38
displaying	38
translations.....	38
Translations.....	38
Translation Environment	38
displaying translator's notes.....	38
displaying translator's notes.....	38
document	38
displaying notes.....	38
displaying notes.....	38
viewing	38
translator's notes.....	38
displaying	38
translator's notes.....	38
segments	38
compressing.....	38
compressing.....	38
segments	38
deleting.....	38
segment	38
expanding.....	38
expanding.....	38
segments	38
joining.....	38
joining.....	38
segments	38
marking.....	38
segments	38
reflowing.....	38
reflowing.....	38
segments	38
spellchecking.....	38
spellchecking.....	38
segments	38
splitting.....	38
splitting.....	38
segments	39
translating.....	39
translating.....	39
segments	39
truncating.....	39
segments	39
untranslating.....	39
translationssaving	39
saving.....	39
translationsprinting	39
printing.....	39
printing	39
document displayed in Translation window	39
document displayed in Translation window.....	39
windowsCustomize Translation Functions	40
options	40
Customize Translation Functions	40
options.....	41
Customize Translation Functions	41
Editor options.....	41
Editor options.....	41
cursorswitching to insert mode	41
switching to insert mode.....	41
editor functions	41
Backspace key.....	41
Backspace key.....	41
Backspace key.....	41
shift-in/shift-out characters, automatic insertion of.....	41
DBCS, automatic insertion of shift-in/shift-out characters.....	41
double-byte characters, automatic insertion of shift-in/shift-out characters.....	41
Customize Translation Functions	41
Display options.....	41
RTFEdit editor	41
visible blank characters	41
visible blank characters.....	41
visible line feed characters.....	41
Customize Translation Functions	42
Directory options.....	42
Customize Translation Functions	42
Translation Memory options.....	42
Translation Memory options.....	42
Translation Memory matches	42
displaying MT matches.....	42
matches	42
displaying MT matches.....	42
Translation Memory matches	42
having terms replaced automatically.....	42
having terms replaced automatically.....	42
matches	42
having terms replaced automatically.....	42
abbreviating Translation Memory proposals	43
abbreviating Translation Memory proposals.....	43
Translation Memory	43
abbreviating proposals.....	43

abbreviating proposals.....	43
abbreviations	
Translation Memory proposals	43
Translation Memory proposals.....	43
editor functions	
display all exact proposals	43
display all exact proposals.....	43
display all	43
proposals	
display all	43
display all.....	43
translation proposals	
display all	43
display all.....	43
Translation Memory	
display all proposals	43
display all proposals.....	43
translation proposal	
quality of proposal	43
quality of proposal.....	43
proposals	
quality of fuzzy matches	43
quality of fuzzy matches.....	43
Translation Memory	
quality of fuzzy matches	43
quality of fuzzy matches.....	43
Customize Translation Functions	
Messages options	43
markup tags	
checking for changes in segments	43
checking for changes in segments.....	43
editor functions	
suppressing messages	43
suppressing messages.....	43
messages	
suppressing	43
suppressing.....	43
suppressing messages.....	43
Customize Translation Functions	
Aut. Substitution options	44
Aut. Substitution options.....	44
automatic substitution	
interrupting	44
interrupting.....	44
fully qualified file names	48
file names, fully qualified.....	48
Document List window	
selecting multiple documents	51
selecting multiple documents.....	51
Folder List window	
selecting multiple folders	51
selecting multiple folders.....	51
Translation Memory List window	
selecting multiple Translation Memories	51
selecting multiple Translation Memories.....	51
Dictionary List window	
selecting multiple dictionaries	51
selecting multiple dictionaries.....	51
Connect Shared Translation Memories window	
selecting multiple shared Translation Memories	51
selecting multiple shared Translation Memories.....	51
Connect Shared Dictionaries window	
selecting multiple shared dictionaries	51
selecting multiple shared dictionaries.....	51
Import Documents window	
selecting multiple documents	51
selecting multiple documents.....	51
Import Folder window	
selecting multiple folders	51
selecting multiple folders.....	51
Import Translation Memory window	
selecting multiple Translation Memories	51
selecting multiple Translation Memories.....	51
Import Dictionary window	
selecting multiple dictionaries	51
selecting multiple dictionaries.....	51
Import New Terms window	
selecting multiple new terms lists	51
selecting multiple new terms lists.....	51
Import Found Terms window	
selecting multiple found terms lists	51
selecting multiple found terms lists.....	51
Import Exclusion Lists window	
selecting multiple exclusion lists	51
selecting multiple exclusion lists.....	51
Import Markup Table List window	
selecting Markup Table List	51
list windows	
arranging list items	54
arranging list items.....	54
filtering list items	55
filtering list items.....	55
properties	
system preferences	59
system preferences.....	59
system preferences, viewing and changing	59
replace generic inline tagging.....	60
replace generic inline tagging.....	60
inline tagging, replace	60
inline tagging, replace	60
Translation Memory	
format-independence	60
Translation Memory	
format-independence	60
format-independence	60
fuzzy match overlap ratio — for translation memory lookup	
default individual settings	60
default	60
individual settings	60

documentformats you can use with TranslationManager	61
formats you can use with TranslationManager	61
layout information in a document.....	62
formatting information in a document.....	62
markup languagedescription	62
description.....	62
analyzingdefinition of	63
definition of.....	63
analyzingdocuments using defaults	63
documents using defaults.....	63
document.....	63
opening.....	63
analyzingdocuments using Translation Memory databases	64
documents using Translation Memory databases	64
untranslated segmentsprocessing them	65
processing them.....	65
untranslated segmentscollecting them in a file	65
collecting them in a file.....	65
untranslated segmentssending them to a machine translation system	65
sending them to a machine translation system.....	65
machine translation system.....	65
analyzingEditing analysis Profiles Translation Memory databases	67
Editing analysis Profiles Translation Memory databases	67
analyzingdocuments using dictionaries	69
documents using dictionaries.....	69
terminology listscreating	69
creating.....	69
terminology listslist of new terms	69
list of new terms.....	69
terminology listslist of found terms	69
list of found terms.....	69
terminology listsexclusion lists	69
exclusion lists.....	69
creatinglists of new terms	70
lists of new terms.....	70
creatinglists of found terms	70
lists of found terms.....	70
abbreviation listmaintaining	71
maintaining	71
documentclosing	72
closing.....	72
stop translating.....	72
ending translations.....	72
saving translations.....	72
leaving the Translation Environment	72
returningto the main window	72
to the main window.....	72
countingwords	72
words.....	72
documentcounting words in	72
counting words in.....	72
documentcounting markup tags in	72
counting markup tags in.....	72
countingmarkup tags in a document	72
markup tags in a document.....	72
countingwords	74
words.....	74
documentcounting duplicate words in	74
counting duplicate words in.....	74
countingmatches	77
matches.....	77
documentcounting matches in	77
counting matches in.....	77
Translation Memorycounting matches from	77
counting matches from.....	77
Translation Memory matchescounting	77
counting.....	77
matchescounting	77
counting.....	77
fuzzy match overlap ratioimpact on counting matches	79
impact on counting matches.....	79
countingmatches	79
matches.....	79
documentCounting words in segments with fuzzy matches	79
Counting words in segments with fuzzy matches.....	79
deletingdocuments	84
documents.....	84
documentdeleting	84

<i>deleting</i>	84
<i>erasing translations</i>	84
<i>exporting documents</i>	84
<i>documents</i>	84
<i>document exporting</i>	84
<i>exporting</i>	84
<i>returning translation to customer</i>	84
<i>translation to customer</i>	84
<i>transferring translation to another user</i>	84
<i>exporting documents in TranslationManager format</i>	85
<i>documents in TranslationManager format</i>	85
<i>document exporting in TranslationManager format</i>	85
<i>exporting in TranslationManager format</i>	85
<i>exporting documents in external format</i>	86
<i>documents in external format</i>	86
<i>document exporting in external format</i>	86
<i>exporting in external format</i>	86
<i>untranslated segments</i>	87
<i>automatic substitution</i>	87
<i>machine translation system</i>	87
<i>exporting documents in external format with path</i>	87
<i>documents in external format with path</i>	87
<i>document exporting in external format with path</i>	87
<i>exporting in external format with path</i>	87
<i>exporting documents in validation format</i>	88
<i>documents in validation format</i>	88
<i>document exporting in validation format</i>	88
<i>exporting in validation format</i>	88
<i>exportingExporting a document in plain XML format</i>	91
<i>Exporting a document in plain XML format</i>	91
<i>documentExporting a document in plain XML format</i>	91
<i>Exporting a document in plain XML format</i>	91
<i>exporting glossary to Lotus Notes</i>	92
<i>glossary to Lotus Notes</i>	92
<i>document exporting glossary to Lotus Notes</i>	92
<i>exporting glossary to Lotus Notes</i>	92
<i>Lotus Notes exporting glossary to</i>	92
<i>exporting glossary to</i>	92
<i>revision marks, defining</i>	93
<i>editing revision marks</i>	93
<i>revision marks</i>	93
<i>document importing</i>	94
<i>importing</i>	94
<i>importing documents</i>	94
<i>documents</i>	94
<i>importing documents in TranslationManager format</i>	95
<i>documents in TranslationManager format</i>	95
<i>document importing in TranslationManager format</i>	95
<i>importing in TranslationManager format</i>	95
<i>importing documents in external format</i>	96
<i>documents in external format</i>	96
<i>document importing in external format</i>	96
<i>importing in external format</i>	96
<i>importing documents in external format with path</i>	98
<i>documents in external format with path</i>	98
<i>document importing in external format with path</i>	98
<i>importing in external format with path</i>	98
<i>importing glossary from Lotus Notes</i>	100
<i>glossary from Lotus Notes</i>	100
<i>document importing glossary from Lotus Notes</i>	100
<i>importing glossary from Lotus Notes</i>	100
<i>Lotus Notes importing glossary from</i>	100
<i>importing glossary from</i>	100
<i>deleting documents</i>	101
<i>documents</i>	101
<i>document copying</i>	101
<i>copying</i>	101
<i>copying documents between folders</i>	101
<i>documents between folders</i>	101
<i>moving documents between folders</i>	101
<i>documents between folders</i>	101
<i>history log file copying documents moving documents</i>	101
<i>copying documents</i>	101
<i>moving documents</i>	101
<i>starting translations</i>	101
<i>translations</i>	101
<i>document opening</i>	101
<i>opening</i>	101

<i>openingdocuments</i>	101
<i>documents</i>	101
<i>documentchecking their spelling</i>	106
<i>checking their spelling</i>	106
<i>spellcheckingdocuments</i>	106
<i>documents</i>	106
<i>spellcheckingaddendum terms, editing</i>	107
<i>addendum terms, editing</i>	107
<i>translatingdocuments</i>	108
<i>documents</i>	108
<i>documenttranslating</i>	108
<i>translating</i>	108
<i>translatingseveral documents</i>	108
<i>several documents</i>	108
<i>documenttranslating several documents</i>	108
<i>translating several documents</i>	108
<i>documentopening several documents</i>	108
<i>opening several documents</i>	108
<i>openingseveral documents</i>	108
<i>several documents</i>	108
<i>documentdisplaying properties</i>	109
<i>displaying properties</i>	109
<i>documentproperties</i>	109
<i>properties</i>	109
<i>documentproperties</i>	109
<i>properties</i>	109
<i>documentchanging properties</i>	110
<i>changing properties</i>	110
<i>dateof source file</i>	111
<i>of source file</i>	111
<i>source filedate of</i>	111
<i>date of</i>	111
<i>statisticsdate of source file</i>	111
<i>date of source file</i>	111
<i>documentproperties</i>	111
<i>properties</i>	111
<i>dateof source file</i>	112
<i>of source file</i>	112
<i>source filedate of</i>	112
<i>date of</i>	112
<i>statisticsdate of source file</i>	112
<i>date of source file</i>	112
<i>documentdetails</i>	112
<i>details</i>	112
<i><TRNOTE> (translator's note) markup</i>	113
<i>translator's notesways to view</i>	113
<i>ways to view</i>	113
<i>viewingtranslator's notes</i>	113
<i>translator's notes</i>	113
<i><TRNOTE> (translator's note) markuhiding shrinking changing name changing color changing font</i>	114
<i>hiding</i>	114
<i>shrinking</i>	114
<i>changing name</i>	114
<i>changing color</i>	114
<i>changing font</i>	114
<i>hiding<TRNOTE> (translator's note) markup</i>	114
<i><TRNOTE> (translator's note) markup</i>	114
<i>changingname of <TRNOTE> (translator's note) markup</i>	114
<i>name of <TRNOTE> (translator's note) markup</i>	114
<i>changingcolor of <TRNOTE> (translator's note) markup</i>	114
<i>color of <TRNOTE> (translator's note) markup</i>	114
<i>changingfont of <TRNOTE> (translator's note) markup</i>	114
<i>font of <TRNOTE> (translator's note) markup</i>	114
<i>RTFEdit editorchanging font of <TRNOTE> markup</i>	114
<i>changing font of <TRNOTE> markup</i>	114
<i>controlled folder handling</i>	115
<i>foldercontrolled folder handling</i>	115
<i>controlled folder handling</i>	115
<i>analyzingall documents in a folder</i>	116
<i>all documents in a folder</i>	116
<i>folderanalyzing the documents it contains</i>	116
<i>analyzing the documents it contains</i>	116
<i>folderclosing</i>	116
<i>closing</i>	116
<i>foldercounting markup tags in the documents it contains</i>	117
<i>counting markup tags in the documents it contains</i>	117
<i>foldercounting words in the documents it contains</i>	117
<i>counting words in the documents it contains</i>	117

<i>foldercreating</i>	117
<i>creating</i>	117
<i>creatingfolders</i>	117
<i>folders</i>	117
<i>folderselecting</i>	119
<i>selecting</i>	119
<i>creatingdictionary</i>	119
<i>dictionary</i>	119
<i>folderselecting</i>	120
<i>selecting</i>	120
<i>creating Translation Memory</i>	120
<i>Translation Memory</i>	120
<i>foldercreating a subfolder</i>	121
<i>creating a subfolder</i>	121
<i>creatingsubfolders</i>	121
<i>subfolders</i>	121
<i>subfoldercreating</i>	121
<i>creating</i>	121
<i>folderrenaming</i>	122
<i>renaming</i>	122
<i>renamingfolders</i>	122
<i>folders</i>	122
<i>folderdeleting</i>	122
<i>deleting</i>	122
<i>deletingfolders</i>	122
<i>folders</i>	122
<i>folderdeleting an exported</i>	123
<i>deleting an exported</i>	123
<i>deletingexported folders</i>	123
<i>exported folders</i>	123
<i>folderelexporting</i>	124
<i>exporting</i>	124
<i>exportingfolders</i>	124
<i>folders</i>	124
<i>importingfolders</i>	126
<i>folders</i>	126
<i>folderimporting</i>	126
<i>importing</i>	126
<i>importingfolders (XLIFF)</i>	128
<i>folders (XLIFF)</i>	128
<i>folderimporting</i>	128
<i>importing</i>	128
<i>folderopening</i>	130
<i>opening</i>	130
<i>openingfolders</i>	130
<i>folders</i>	130
<i>foldersearching and replacing text in its documents</i>	130
<i>searching and replacing text in its documents</i>	130
<i>folderdisplaying properties</i>	131
<i>displaying properties</i>	131
<i>folderproperties</i>	131
<i>properties</i>	131
<i>folderchanging properties</i>	132
<i>changing properties</i>	132
<i>folderproperties</i>	132
<i>properties</i>	132
<i>folderdetails</i>	134
<i>details</i>	134
<i>Translation Memorydefinition of</i>	135
<i>definition of</i>	135
<i>Translation Memory matchesexact</i>	135
<i>exact</i>	135
<i>matchesexact</i>	135
<i>exact</i>	135
<i>Translation Memory matchesfuzzy</i>	135
<i>fuzzy</i>	135
<i>matchesfuzzy</i>	135
<i>fuzzy</i>	135
<i>Translation Memory matchestypes of</i>	136
<i>types of</i>	136
<i>matchestypes of</i>	136
<i>types of</i>	136
<i>fuzzy match overlap ratiodefault</i>	136
<i>default</i>	136
<i>Translation Memory matchescategories</i>	136
<i>categories</i>	136
<i>matchescategories</i>	136
<i>categories</i>	136

<i>creating Translation Memory databases</i>	137
<i>Translation Memory databases</i>	137
<i>Translation Memory</i>	137
<i>creating</i>	137
<i>renaming</i>	141
<i>renaming Translation Memory databases</i>	141
<i>Translation Memory databases</i>	141
<i>deleting</i>	141
<i>deleting Translation Memory databases</i>	141
<i>Translation Memory databases</i>	141
<i>exporting</i>	142
<i>exporting Translation Memory databases</i>	142
<i>Translation Memory databases</i>	142
<i>Translation Memory</i>	142
<i>Unicode enablement</i>	142
<i>Unicode enablement</i>	142
<i>reusing existing translations</i>	143
<i>existing translations</i>	143
<i>Initial Translation Memory (ITM)</i>	143
<i>creating</i>	143
<i>creating Initial Translation Memory (ITM) databases</i>	143
<i>Initial Translation Memory (ITM) databases</i>	143
<i>commandsEQFITM</i>	143
<i>EQFITM</i>	143
<i>translationsreusing existing ones</i>	143
<i>reusing existing ones</i>	143
<i>Translation Memory matches</i>	143
<i>machine-generated</i>	143
<i>machine-generated</i>	143
<i>matches</i>	143
<i>machine-generated</i>	143
<i>importing Translation Memory databases</i>	146
<i>Translation Memory databases</i>	146
<i>Translation Memory</i>	146
<i>importing</i>	146
<i>importing</i>	146
<i>Translation Memory</i>	146
<i>Unicode enablement</i>	146
<i>Unicode enablement</i>	146
<i>Translation Memory</i>	147
<i>merging</i>	147
<i>merging</i>	147
<i>merging Translation Memory databases</i>	147
<i>Translation Memory databases</i>	147
<i>Translation Memory</i>	148
<i>archiving</i>	148
<i>archiving</i>	148
<i>archiving Translation Memory databases</i>	148
<i>Translation Memory</i>	149
<i>organizing</i>	149
<i>organizing</i>	149
<i>organizing Translation Memory databases</i>	149
<i>Translation Memory databases</i>	149
<i>Translation Memory</i>	149
<i>removing obsolete segments</i>	149
<i>removing obsolete segments</i>	149
<i>Translation Memory</i>	149
<i>repairing defective ones</i>	149
<i>repairing defective ones</i>	149
<i>Translation Memory</i>	149
<i>upgrading backlevel versions</i>	149
<i>upgrading backlevel versions</i>	149
<i>revising</i>	149
<i>Initial Translation Memory</i>	149
<i>Initial Translation Memory</i>	149
<i>postediting</i>	149
<i>Initial Translation Memory</i>	149
<i>Initial Translation Memory</i>	149
<i>Initial Translation Memory (ITM)</i>	149
<i>revising</i>	149
<i>revising</i>	149
<i>Initial Translation Memory (ITM)</i>	149
<i>postediting</i>	149
<i>postediting</i>	149
<i>editing</i>	149
<i>Initial Translation Memory</i>	149
<i>Initial Translation Memory</i>	149
<i>revising</i>	149
<i>Translation Memory databases</i>	153
<i>Translation Memory databases</i>	153
<i>postediting</i>	153
<i>Translation Memory databases</i>	153
<i>Translation Memory databases</i>	153
<i>Translation Memory</i>	153
<i>revising</i>	153
<i>Translation Memory</i>	153
<i>postediting</i>	153
<i>editing</i>	153
<i>Translation Memory databases</i>	153
<i>Translation Memory databases</i>	153
<i>Translation Memory</i>	154
<i>searching a translation</i>	154
<i>searching a translation</i>	154
<i>Translation Memory</i>	159
<i>displaying properties</i>	159

<i>displaying properties</i>	159
<i>Translation Memory</i> <i>properties</i>	159
<i>changing properties</i>	161
<i>changing properties</i>	161
<i>dictionaryproperties</i>	164
<i>properties</i>	164
<i>dictionarysharing</i>	165
<i>sharing</i>	165
<i>sharingdictionaries</i>	165
<i>dictionaries</i>	165
<i>dictionarypersonal</i>	165
<i>personal</i>	165
<i>dictionarydefault entry fields</i>	170
<i>default entry fields</i>	170
<i>dictionarychanging entry fields</i>	172
<i>changing entry fields</i>	172
<i>dictionaryrenaming entry fields</i>	174
<i>renaming entry fields</i>	174
<i>dictionaryadding your own entry fields</i>	174
<i>adding your own entry fields</i>	174
<i>dictionary filter</i>	175
<i>filterdefinition of</i>	175
<i>definition of</i>	175
<i>filterfor printing dictionary extracts</i>	175
<i>for printing dictionary extracts</i>	175
<i>filterfor looking up dictionary entries</i>	175
<i>for looking up dictionary entries</i>	175
<i>conditions</i>	175
<i>dictionaryrenaming</i>	177
<i>renaming</i>	177
<i>renamingdictionaries</i>	177
<i>dictionaries</i>	177
<i>dictionarydeleting</i>	178
<i>deleting</i>	178
<i>deletingdictionaries</i>	178
<i>dictionaries</i>	178
<i>dictionaryupdating an entry</i>	178
<i>updating an entry</i>	178
<i>dictionaryadding an entry</i>	178
<i>adding an entry</i>	178
<i>dictionaryremoving an entry</i>	178
<i>removing an entry</i>	178
<i>dictionarytemplate</i>	179
<i>template</i>	179
<i>dictionaryexporting</i>	180
<i>exporting</i>	180
<i>exportingdictionaries</i>	180
<i>dictionaries</i>	180
<i>dictionaryUnicode enablement</i>	180
<i>Unicode enablement</i>	180
<i>dictionaryimporting</i>	181
<i>importing</i>	181
<i>importingdictionaries</i>	181
<i>dictionaries</i>	181
<i>dictionarymerging when importing</i>	181
<i>merging when importing</i>	181
<i>dictionaryUnicode enablement</i>	181
<i>Unicode enablement</i>	181
<i>dictionarylooking up an entry</i>	182
<i>looking up an entry</i>	182
<i>mergingdictionaries</i>	184
<i>dictionaries</i>	184
<i>dictionarymerging</i>	184
<i>merging</i>	184
<i>dictionaryorganizing</i>	185
<i>organizing</i>	185
<i>organizingdictionaries</i>	185
<i>dictionaries</i>	185
<i>dictionaryprinting</i>	186
<i>printing</i>	186
<i>printingdictionary extracts</i>	186
<i>dictionary extracts</i>	186
<i>dictionaryprotecting from uncontrolled modifications</i>	188
<i>protecting from uncontrolled modifications</i>	188
<i>protecting dictionaries</i>	188
<i>dictionaries</i>	188
<i>dictionaryassigning a password</i>	188

<i>assigning a password</i>	188
<i>searching for a term in a dictionary</i>	189
<i>dictionary</i> <i>searching for a term</i>	189
<i>searching for a term</i>	189
<i>dictionary</i> <i>displaying properties</i>	191
<i>displaying properties</i>	191
<i>dictionary</i> <i>properties</i>	191
<i>properties</i>	191
<i>dictionary</i> <i>displaying properties</i>	192
<i>displaying properties</i>	192
<i>dictionary</i> <i>changing properties</i>	192
<i>changing properties</i>	192
<i>dictionary</i> <i>properties</i>	192
<i>properties</i>	192
<i>new terms list</i>	196
<i>found terms list</i>	196
<i>exclusion list</i>	196
<i>terminology lists</i> <i>creating lists of new terms</i>	197
<i>creating lists of new terms</i>	197
<i>terminology lists</i> <i>creating lists of found terms</i>	197
<i>creating lists of found terms</i>	197
<i>creating lists of found terms</i>	197
<i>lists of found terms</i>	197
<i>creating lists of new terms</i>	197
<i>lists of new terms</i>	197
<i>new terms list</i> <i>creating</i>	197
<i>creating</i>	197
<i>found terms list</i> <i>creating</i>	197
<i>creating</i>	197
<i>creating exclusion lists</i>	197
<i>exclusion lists</i>	197
<i>terminology lists</i> <i>creating exclusion lists</i>	197
<i>creating exclusion lists</i>	197
<i>exclusion lists</i> <i>creating</i>	197
<i>creating</i>	197
<i>terminology lists</i> <i>removing</i>	198
<i>removing</i>	198
<i>deleting terminology lists</i>	198
<i>terminology lists</i>	198
<i>exporting terminology lists</i>	199
<i>terminology lists</i>	199
<i>terminology list</i> <i>exporting</i>	199
<i>exporting</i>	199
<i>importing terminology lists</i>	200
<i>terminology lists</i>	200
<i>terminology lists</i> <i>importing</i>	200
<i>importing</i>	200
<i>terminology lists</i> <i>modifying</i>	201
<i>modifying</i>	201
<i>terminology lists</i> <i>moving terms to another list</i>	201
<i>moving terms to another list</i>	201
<i>opening terminology lists</i>	201
<i>terminology lists</i>	201
<i>terminology lists</i> <i>moving terms to dictionary</i>	203
<i>moving terms to dictionary</i>	203
<i>terminology lists</i> <i>marking terms for adding to dictionary</i>	203
<i>marking terms for adding to dictionary</i>	203
<i>terminology lists</i> <i>modifying an exclusion list</i>	204
<i>modifying an exclusion list</i>	204
<i>exclusion lists</i> <i>adding a term</i>	204
<i>adding a term</i>	204
<i>exclusion lists</i> <i>changing a term</i>	204
<i>changing a term</i>	204
<i>exclusion lists</i> <i>removing a term</i>	204
<i>removing a term</i>	204
<i>editing exclusion lists</i>	204
<i>exclusion lists</i>	204
<i>terminology lists</i> <i>printing</i>	204
<i>printing</i>	204
<i>printing terminology lists</i>	204
<i>terminology lists</i>	204
<i>markup tables</i> <i>delivered with TranslationManager</i>	205
<i>delivered with TranslationManager</i>	205
<i>markup table</i> <i>exporting</i> <i>importing</i>	206
<i>importing</i>	206
<i>changing properties</i>	206
<i>deleting markup tables</i>	207

<i>markup tables</i>	207
<i>markup tables</i> removing	207
removing	207
<i>exporting</i> <i>markup tables</i>	208
<i>markup tables</i>	208
<i>markup tablesexporting</i>	208
<i>exporting</i>	208
<i>markup tables</i> <i>importing</i>	209
<i>importing</i>	209
<i>markup tables</i> <i>making them available</i>	209
<i>making them available</i>	209
<i>importing</i> <i>markup tables</i>	209
<i>markup tables</i>	209
<i>markup tables</i> <i>changing properties</i>	210
<i>changing properties</i>	210
<i>changing properties</i> <i>of markup tables</i>	210
<i>of markup tables</i>	210
<i>properties</i> <i>of markup tables</i>	210
<i>of markup tables</i>	210
<i>windows</i> <i>Markup Table Properties window</i>	211
<i>Markup Table Properties window</i>	211
<i>Markup Table Properties window</i>	211
<i>Markup Table Properties window</i> <i>General page</i>	211
<i>General page</i>	211
<i>markup tables</i> <i>description of</i>	211
<i>description of</i>	211
<i>markup tables</i> <i>descriptive name of</i>	211
<i>descriptive name of</i>	211
<i>markup tables</i> <i>user exit</i>	211
<i>user exit</i>	211
<i>markup tables</i> <i>character set used with</i>	211
<i>character set used with</i>	211
<i>markup table</i> <i>substitution character used in</i>	211
<i>substitution character used in</i>	211
<i>markup table</i> <i>substitution character used in</i>	212
<i>substitution character used in</i>	212
<i>markup table</i> <i>substitution character used in</i>	212
<i>substitution character used in</i>	212
<i>markup table</i> <i>translator's notes in</i>	212
<i>translator's notes in</i>	212
<i>markup table</i> <i>translator's note</i> <i>start tag</i>	212
<i>translator's note</i> <i>start tag</i>	212
<i>start tag</i>	212
<i>markup table</i> <i>translator's note</i> <i>level 1 note</i>	212
<i>translator's note</i> <i>level 1 note</i>	212
<i>level 1 note</i>	212
<i>translator's note</i> <i>level 1</i>	212
<i>level 1</i>	212
<i>markup table</i> <i>translator's note</i> <i>level 2 note</i>	212
<i>translator's note</i> <i>level 2 note</i>	212
<i>level 2 note</i>	212
<i>translator's note</i> <i>level 2</i>	212
<i>level 2</i>	212
<i>markup table</i> <i>translator's note</i> <i>end tag</i>	212
<i>translator's note</i> <i>end tag</i>	212
<i>end tag</i>	212
<i>Markup Table Properties window</i> <i>Tags page</i>	212
<i>Tags page</i>	212
<i>markup table</i> <i>tag definitions in</i>	212
<i>tag definitions in</i>	212
<i>Markup Table Properties window</i> <i>Attributes page</i>	213
<i>Attributes page</i>	213
<i>markup table</i> <i>attribute definitions in</i>	213
<i>attribute definitions in</i>	213
<i>markup table</i> <i>testing</i>	214
<i>testing</i>	214
<i>testing</i> <i>markup tables</i>	214
<i>markup tables</i>	214
<i>Markup Table Properties window</i> <i>Test page</i>	214
<i>Test page</i>	214
<i>markup table</i> <i>protecting</i>	214
<i>protecting</i>	214
<i>protecting</i> <i>markup tables</i>	214
<i>markup tables</i>	214
<i>stem reduction</i>	215
<i>dictionary</i> <i>stem reduction during lookup</i>	215
<i>stem reduction during lookup</i>	215
<i>compound separation</i>	215

dictionary	compound separation during lookup	215
	compound separation during lookup.....	215
language-support files	deleting	215
	deleting.....	215
deleting	language-support files	215
	language-support files.....	215
sample material	importing sample folders and documents	217
	importing sample folders and documents.....	217
importing	sample folders and documents	217
	sample folders and documents.....	217
openingsample	folders and documents	217
	sample folders and documents.....	217
reportsHistory	Report	221
	History Report.....	221
History	Report.....	221
reportsCounting	Report	223
	Counting Report.....	223
Counting	Report.....	223
fuzzy	match overlap ratioimpact on counting report	224
	impact on counting report.....	224
reportsCalculating	Report	226
	Calculating Report.....	226
Calculating	Report.....	226
reportsPreanalysis	Report	230
	Preanalysis Report.....	230
Preanalysis	Report.....	230
reportsRedundancy	Report	233
	Redundancy Report.....	233
Redundancy	Report.....	233
reportsRedundant	Segment List	236
	Redundant Segment List.....	236
Redundant	Segment List.....	236
documentanalyzing	(from command area)	246
	analyzing (from command area).....	246
analyzingdocuments	(from command area)	246
	documents (from command area).....	246
EQFCMDanalyzing	documents	246
	analyzing documents.....	246
commandsEQFCMDanalyzing	a document or folder	246
	EQFCMDanalyzing a document or folder	246
EQFCMDanalyzing	a document or folder	246
	analyzing a document or folder.....	246
documentexporting	(from command area)	248
	exporting (from command area).....	248
exportingdocuments	(from command area)	248
	documents (from command area).....	248
EQFCMDexporting	documents	248
	exporting documents.....	248
commandsEQFCMDexporting	documents	248
	EQFCMDexporting documents	248
EQFCMDexporting	documents	248
	exporting documents.....	248
documentimporting	(from command area)	250
	importing (from command area).....	250
importingdocuments	(from command area)	250
	documents (from command area).....	250
EQFCMDimporting	documents	250
	importing documents.....	250
commandsEQFCMDimporting	documents	250
	EQFCMDimporting documents	250
EQFCMDimporting	documents	250
	importing documents.....	250
documentdeleting	(from command area)	252
	deleting (from command area).....	252
deletingdocuments	(from command area)	252
	documents (from command area).....	252
EQFCMDdeleting	documents	252
	deleting documents.....	252
commandsEQFCMDdeleting	documents	252
	EQFCMDdeleting documents	252
EQFCMDdeleting	documents	252
	deleting documents.....	252
foldercreating	(from command area)	253
	creating (from command area).....	253
creatingfolders	(from command area)	253
	folders (from command area)	253
EQFCMDcreating	folders	253
	creating folders.....	253
commandsEQFCMDcreating	folders	253
	EQFCMDcreating folders	253
EQFCMDcreating	folders	253
	creating folders.....	253
folderexporting	(from command area)	254

<i>exporting (from command area)</i>	254
<i>exportingfolders (from command area)</i>	254
<i>folders (from command area)</i>	254
<i>EQFCMD</i> <i>exporting folders</i>	254
<i>exporting folders</i>	254
<i>commands</i> <i>EQFCMD</i> <i>exporting folders</i>	254
<i>EQFCMD</i> <i>exporting folders</i>	254
<i>exporting folders</i>	254
<i>folder</i> <i>importing (from command area)</i>	256
<i>importing (from command area)</i>	256
<i>importingfolders (from command area)</i>	256
<i>folders (from command area)</i>	256
<i>EQFCMD</i> <i>importing folders</i>	256
<i>importing folders</i>	256
<i>commands</i> <i>EQFCMD</i> <i>importing folders</i>	256
<i>EQFCMD</i> <i>importing folders</i>	256
<i>importing folders</i>	256
<i>folder</i> <i>deleting (from command area)</i>	257
<i>deleting (from command area)</i>	257
<i>deletingfolders (from command area)</i>	257
<i>folders (from command area)</i>	257
<i>EQFCMD</i> <i>deleting folders</i>	257
<i>deleting folders</i>	257
<i>commands</i> <i>EQFCMD</i> <i>deleting folders</i>	257
<i>EQFCMD</i> <i>deleting folders</i>	257
<i>deleting folders</i>	257
<i>Translation Memory</i> <i>creating (from command area)</i>	257
<i>creating (from command area)</i>	257
<i>creating</i> <i>Translation Memory databases (from command area)</i>	257
<i>Translation Memory databases (from command area)</i>	257
<i>EQFCMD</i> <i>creating Translation Memory databases</i>	257
<i>creating Translation Memory databases</i>	257
<i>commands</i> <i>EQFCMD</i> <i>creating Translation Memory databases</i>	257
<i>EQFCMD</i> <i>creating Translation Memory databases</i>	257
<i>creating Translation Memory databases</i>	257
<i>Translation Memory</i> <i>deleting (from command area)</i>	258
<i>deleting (from command area)</i>	258
<i>deleting</i> <i>Translation Memory databases (from command area)</i>	258
<i>Translation Memory databases (from command area)</i>	258
<i>EQFCMD</i> <i>deleting Translation Memory databases</i>	258
<i>deleting Translation Memory databases</i>	258
<i>commands</i> <i>EQFCMD</i> <i>deleting Translation Memory databases</i>	258
<i>EQFCMD</i> <i>deleting Translation Memory databases</i>	258
<i>deleting Translation Memory databases</i>	258
<i>Translation Memory</i> <i>exporting (from command area)</i>	259
<i>exporting (from command area)</i>	259
<i>exporting</i> <i>Translation Memory databases (from command area)</i>	259
<i>Translation Memory databases (from command area)</i>	259
<i>EQFCMD</i> <i>exporting Translation Memory databases</i>	259
<i>exporting Translation Memory databases</i>	259
<i>commands</i> <i>EQFCMD</i> <i>exporting Translation Memory databases</i>	259
<i>EQFCMD</i> <i>exporting Translation Memory databases</i>	259
<i>exporting Translation Memory databases</i>	259
<i>Translation Memory</i> <i>importing (from command area)</i>	260
<i>importing (from command area)</i>	260
<i>importing</i> <i>Translation Memory databases (from command area)</i>	260
<i>Translation Memory databases (from command area)</i>	260
<i>EQFCMD</i> <i>importing Translation Memory databases</i>	260
<i>importing Translation Memory databases</i>	260
<i>commands</i> <i>EQFCMD</i> <i>importing Translation Memory databases</i>	260
<i>EQFCMD</i> <i>importing Translation Memory databases</i>	260
<i>importing Translation Memory databases</i>	260
<i>dictionary</i> <i>exporting (from command area)</i>	260
<i>exporting (from command area)</i>	260
<i>exporting</i> <i>dictionaries (from command area)</i>	260
<i>dictionaries (from command area)</i>	260
<i>EQFCMD</i> <i>exporting dictionaries</i>	260
<i>exporting dictionaries</i>	260
<i>commands</i> <i>EQFCMD</i> <i>exporting dictionaries</i>	260
<i>EQFCMD</i> <i>exporting dictionaries</i>	260
<i>exporting dictionaries</i>	260
<i>dictionary</i> <i>importing (from command area)</i>	261
<i>importing (from command area)</i>	261
<i>importing</i> <i>dictionaries (from command area)</i>	261
<i>dictionaries (from command area)</i>	261
<i>EQFCMD</i> <i>importing dictionaries</i>	261
<i>importing dictionaries</i>	261
<i>commands</i> <i>EQFCMD</i> <i>importing dictionaries</i>	261

<i>EQFCMD</i>	<i>Importing dictionaries</i>	261
	<i>importing dictionaries</i>	261
<i>archiving Translation Memory databases from command area</i>	262	
	<i>from command area</i>	262
<i>Translation Memory</i>	<i>archiving (from command area)</i>	262
	<i>archiving (from command area)</i>	262
<i>EQFCMD</i>	<i>Archiving Translation Memory databases</i>	262
	<i>archiving Translation Memory databases</i>	262
<i>commands</i>	<i>EQFCMD</i> <i>Archiving Translation Memory databases</i>	262
	<i>EQFCMD</i> <i>Archiving Translation Memory databases</i>	262
	<i>archiving Translation Memory databases</i>	262
<i>organizing Translation Memory databases (from command area)</i>	263	
	<i>Translation Memory databases (from command area)</i>	263
<i>Translation Memory</i>	<i>organizing (from command area)</i>	263
	<i>organizing (from command area)</i>	263
<i>EQFCMD</i>	<i>Organizing Translation Memory databases</i>	263
	<i>organizing Translation Memory databases</i>	263
<i>commands</i>	<i>EQFCMD</i> <i>Organizing Translation Memory databases</i>	263
	<i>EQFCMD</i> <i>Organizing Translation Memory databases</i>	263
	<i>organizing Translation Memory databases</i>	263
<i>document</i>	<i>opening (from command area)</i>	263
	<i>opening (from command area)</i>	263
<i>opening</i>	<i>documents (from command area)</i>	263
	<i>documents (from command area)</i>	263
<i>EQFCMD</i>	<i>Opening documents</i>	264
	<i>opening documents</i>	264
<i>commands</i>	<i>EQFCMD</i> <i>Opening documents</i>	264
	<i>EQFCMD</i> <i>Opening documents</i>	264
	<i>opening documents</i>	264
<i>counting</i>	<i>words (from command area)</i>	264
	<i>words (from command area)</i>	264
<i>EQFCMD</i>	<i>Counting words</i>	264
	<i>counting words</i>	264
<i>commands</i>	<i>EQFCMD</i> <i>Counting words</i>	264
	<i>EQFCMD</i> <i>Counting words</i>	264
	<i>counting words</i>	264
<i>creating</i>	<i>reports (from command area)</i>	265
	<i>reports (from command area)</i>	265
<i>EQFCMD</i>	<i>Creating reports</i>	265
	<i>creating reports</i>	265
<i>commands</i>	<i>EQFCMD</i> <i>Creating reports</i>	265
	<i>EQFCMD</i> <i>Creating reports</i>	265
	<i>creating reports</i>	265
<i>document</i>	<i>EQFBATCH: analyzing (from command area)</i>	269
	<i>EQFBATCH: analyzing (from command area)</i>	269
<i>analyzing</i>	<i>EQFBATCH: documents (from command area)</i>	269
	<i>EQFBATCH: documents (from command area)</i>	269
<i>EQFBATCH</i>	<i>Analyzing documents</i>	269
	<i>analyzing documents</i>	269
<i>commands</i>	<i>EQFBATCH</i> <i>Analyzing a document or folder</i>	269
	<i>EQFBATCH</i> <i>Analyzing a document or folder</i>	269
	<i>analyzing a document or folder</i>	269
<i>archiving</i>	<i>Translation Memory databases from command area (EQFBATCH)</i>	270
	<i>from command area (EQFBATCH)</i>	270
<i>Translation Memory</i>	<i>EQFBATCH: archiving (from command area)</i>	270
	<i>EQFBATCH: archiving (from command area)</i>	270
<i>EQFBATCH</i>	<i>Archiving Translation Memory databases</i>	270
	<i>archiving Translation Memory databases</i>	270
<i>commands</i>	<i>EQFBATCH</i> <i>Archiving Translation Memory databases</i>	270
	<i>EQFBATCH</i> <i>Archiving Translation Memory databases</i>	270
	<i>archiving Translation Memory databases</i>	270
<i>creating</i>	<i>EQFBATCH: reports (from command area)</i>	271
	<i>EQFBATCH: reports (from command area)</i>	271
<i>EQFBATCH</i>	<i>Creating reports</i>	271
	<i>creating reports</i>	271
<i>commands</i>	<i>EQFBATCH</i> <i>Creating reports</i>	271
	<i>EQFBATCH</i> <i>Creating reports</i>	271
	<i>creating reports</i>	271
<i>counting</i>	<i>EQFBATCH: words (from command area)</i>	273
	<i>EQFBATCH: words (from command area)</i>	273
<i>EQFBATCH</i>	<i>Counting words</i>	273
	<i>counting words</i>	273
<i>commands</i>	<i>EQFBATCH</i> <i>Counting words</i>	273
	<i>EQFBATCH</i> <i>Counting words</i>	273
	<i>counting words</i>	273
<i>dictionary</i>	<i>EQFBATCH: exporting (from command area)</i>	274
	<i>EQFBATCH: exporting (from command area)</i>	274
<i>exporting</i>	<i>EQFBATCH: dictionaries (from command area)</i>	274

<i>EQFBATCH: dictionaries (from command area)</i>	274
<i>EQFBATCH</i> exporting dictionaries	274
<i>exporting dictionaries</i>	274
<i>commandsEQFBATCH</i> exporting dictionaries	274
<i>EQFBATCH</i> exporting dictionaries	274
<i>exporting dictionaries</i>	274
<i>dictionaryEQFBATCH: importing (from command area)</i>	275
<i>EQFBATCH: importing (from command area)</i>	275
<i>importingEQFBATCH: dictionaries (from command area)</i>	275
<i>EQFBATCH: dictionaries (from command area)</i>	275
<i>EQFBATCH</i> importing dictionaries	275
<i>importing dictionaries</i>	275
<i>commandsEQFBATCH</i> importing dictionaries	275
<i>EQFBATCH</i> importing dictionaries	275
<i>importing dictionaries</i>	275
<i>documentEQFBATCH: deleting (from command area)</i>	276
<i>EQFBATCH: deleting (from command area)</i>	276
<i>deletingEQFBATCH: documents (from command area)</i>	276
<i>EQFBATCH: documents (from command area)</i>	276
<i>EQFBATCH</i> deleting documents	276
<i>deleting documents</i>	276
<i>commandsEQFBATCH</i> deleting documents	276
<i>EQFBATCH</i> deleting documents	276
<i>deleting documents</i>	276
<i>documentEQFBATCH: exporting (from command area)</i>	277
<i>EQFBATCH: exporting (from command area)</i>	277
<i>exportingEQFBATCH: documents (from command area)</i>	277
<i>EQFBATCH: documents (from command area)</i>	277
<i>EQFBATCH</i> exporting documents	277
<i>exporting documents</i>	277
<i>commandsEQFBATCH</i> exporting documents	277
<i>EQFBATCH</i> exporting documents	277
<i>exporting documents</i>	277
<i>documentEQFBATCH: importing (from command area)</i>	279
<i>EQFBATCH: importing (from command area)</i>	279
<i>importingEQFBATCH: documents (from command area)</i>	279
<i>EQFBATCH: documents (from command area)</i>	279
<i>EQFBATCH</i> importing documents	279
<i>importing documents</i>	279
<i>commandsEQFBATCH</i> importing documents	279
<i>EQFBATCH</i> importing documents	279
<i>importing documents</i>	279
<i>folderEQFBATCH: creating (from command area)</i>	282
<i>EQFBATCH: creating (from command area)</i>	282
<i>creatingEQFBATCH: folders (from command area)</i>	282
<i>EQFBATCH: folders (from command area)</i>	282
<i>EQFBATCH</i> creating folders	282
<i>creating folders</i>	282
<i>commandsEQFBATCH</i> creating folders	282
<i>EQFBATCH</i> creating folders	282
<i>creating folders</i>	282
<i>folderEQFBATCH: deleting (from command area)</i>	283
<i>EQFBATCH: deleting (from command area)</i>	283
<i>deletingEQFBATCH: folders (from command area)</i>	283
<i>EQFBATCH: folders (from command area)</i>	283
<i>EQFBATCH</i> deleting folders	283
<i>deleting folders</i>	283
<i>commandsEQFBATCH</i> deleting folders	283
<i>EQFBATCH</i> deleting folders	283
<i>deleting folders</i>	283
<i>folderEQFBATCH: exporting (from command area)</i>	283
<i>EQFBATCH: exporting (from command area)</i>	283
<i>exportingEQFBATCH: folders (from command area)</i>	283
<i>EQFBATCH: folders (from command area)</i>	283
<i>EQFBATCH</i> exporting folders	284
<i>exporting folders</i>	284
<i>commandsEQFBATCH</i> exporting folders	284
<i>EQFBATCH</i> exporting folders	284
<i>exporting folders</i>	284
<i>folderEQFBATCH: importing (from command area)</i>	285
<i>EQFBATCH: importing (from command area)</i>	285
<i>importingEQFBATCH: folders (from command area)</i>	285
<i>EQFBATCH: folders (from command area)</i>	285
<i>EQFBATCH</i> importing folders	285
<i>importing folders</i>	285
<i>commandsEQFBATCH</i> importing folders	285
<i>EQFBATCH</i> importing folders	285
<i>importing folders</i>	285

<i>Translation MemoryEQFBATCH: creating (from command area)</i>	286
<i>EQFBATCH: creating (from command area)</i>	286
<i>creatingEQFBATCH: Translation Memory databases (from command area)</i>	286
<i>EQFBATCH: Translation Memory databases (from command area)</i>	286
<i>EQFBATCHcreating Translation Memory databases</i>	286
<i>creating Translation Memory databases</i>	286
<i>commandsEQFBATCHcreating Translation Memory databases</i>	286
<i>EQFBATCHcreating Translation Memory databases</i>	286
<i>creating Translation Memory databases</i>	286
<i>Translation MemoryEQFBATCH: deleting (from command area)</i>	287
<i>EQFBATCH: deleting (from command area)</i>	287
<i>deletingEQFBATCH: Translation Memory databases (from command area)</i>	287
<i>EQFBATCH: Translation Memory databases (from command area)</i>	287
<i>EQFBATCHdeleting Translation Memory databases</i>	287
<i>deleting Translation Memory databases</i>	287
<i>commandsEQFBATCHdeleting Translation Memory databases</i>	287
<i>EQFBATCHdeleting Translation Memory databases</i>	287
<i>deleting Translation Memory databases</i>	287
<i>Translation MemoryEQFBATCH: exporting (from command area)</i>	288
<i>EQFBATCH: exporting (from command area)</i>	288
<i>exportingEQFBATCH: Translation Memory databases (from command area)</i>	288
<i>EQFBATCH: Translation Memory databases (from command area)</i>	288
<i>EQFBATCHexporting Translation Memory databases</i>	288
<i>exporting Translation Memory databases</i>	288
<i>commandsEQFBATCHexporting Translation Memory databases</i>	288
<i>EQFBATCHexporting Translation Memory databases</i>	288
<i>exporting Translation Memory databases</i>	288
<i>Translation MemoryEQFBATCH: importing (from command area)</i>	289
<i>EQFBATCH: importing (from command area)</i>	289
<i>importingEQFBATCH: Translation Memory databases (from command area)</i>	289
<i>EQFBATCH: Translation Memory databases (from command area)</i>	289
<i>EQFBATCHimporting Translation Memory databases</i>	289
<i>importing Translation Memory databases</i>	289
<i>commandsEQFBATCHimporting Translation Memory databases</i>	289
<i>EQFBATCHimporting Translation Memory databases</i>	289
<i>importing Translation Memory databases</i>	289
<i>organizingEQFBATCH: Opening a document</i>	290
<i>EQFBATCH: Opening a document</i>	290
<i>Translation MemoryEQFBATCH: Opening a document</i>	290
<i>EQFBATCH: Opening a document</i>	290
<i>EQFBATCHorganizing Translation Memory databases</i>	290
<i>organizing Translation Memory databases</i>	290
<i>commandsEQFBATCHorganizing Translation Memory databases</i>	290
<i>EQFBATCHorganizing Translation Memory databases</i>	290
<i>organizing Translation Memory databases</i>	290
<i>organizingEQFBATCH: Translation Memory databases (from command area)</i>	290
<i>EQFBATCH: Translation Memory databases (from command area)</i>	290
<i>Translation MemoryEQFBATCH: organizing (from command area)</i>	290
<i>EQFBATCH: organizing (from command area)</i>	290
<i>EQFBATCHorganizing Translation Memory databases</i>	291
<i>organizing Translation Memory databases</i>	291
<i>commandsEQFBATCHorganizing Translation Memory databases</i>	291
<i>EQFBATCHorganizing Translation Memory databases</i>	291
<i>organizing Translation Memory databases</i>	291
<i>organizingEQFBATCH: Renaming a folder, a dictionary or a Translation Memory (from command area)</i>	291
<i>EQFBATCH: Renaming a folder, a dictionary or a Translation Memory (from command area)</i>	291
<i>Translation MemoryEQFBATCH: rename (from command area)</i>	291
<i>EQFBATCH: rename (from command area)</i>	291
<i>EQFBATCHrename a folder, a dictionary or a Translation Memory</i>	291
<i>rename a folder, a dictionary or a Translation Memory</i>	291
<i>commandsEQFBATCHrename a folder, a dictionary or a Translation Memory</i>	291
<i>EQFBATCHrename a folder, a dictionary or a Translation Memory</i>	291
<i>rename a folder, a dictionary or a Translation Memory</i>	291
<i>Initial Translation Memory (ITM)creating (from command area)</i>	294
<i>creating (from command area)</i>	294
<i>creatingInitial Translation Memory (ITM) databases (from command area)</i>	294
<i>Initial Translation Memory (ITM) databases (from command area)</i>	294
<i>EQFITMcreating an Initial Translation Memory</i>	295
<i>creating an Initial Translation Memory</i>	295
<i>commandsEQFITMcreating an Initial Translation Memory</i>	295
<i>EQFITMcreating an Initial Translation Memory</i>	295
<i>creating an Initial Translation Memory</i>	295
<i>Initial Translation Memory (ITM)checking segment pairs</i>	297
<i>checking segment pairs</i>	297
<i>Translation Memory matchesmachine-generated</i>	297
<i>machine-generated</i>	297
<i>matchesmachine-generated</i>	297
<i>machine-generated</i>	297

<i>Initial Translation Memory (ITM)checking correctness of matches</i>	297
<i> checking correctness of matches</i>	297
<i>commandsEQFDMM</i>	297
<i> EQFDMM</i>	297
<i> EQFDMM</i>	297
<i> Translation Memoryreversing</i>	298
<i> reversing</i>	298
<i> EQFREVM</i>	298
<i> commandsEQFREVM</i>	298
<i> EQFREVM</i>	298
<i> Translation Memorychanging m-flagged segments (from command area)</i>	298
<i> changing m-flagged segments (from command area)</i>	298
<i> segmentschanging m-flagged segments (from command area)</i>	298
<i> changing m-flagged segments (from command area)</i>	298
<i> EQFCMM</i>	298
<i> commandsEQFCMM</i>	298
<i> EQFCMM</i>	298
<i> Initial Translation Memory (ITM)specifying the quality of m-flagged segments (from command area)</i>	299
<i> specifying the quality of m-flagged segments (from command area)</i>	299
<i> segmentsspecifying the quality of m-flagged segments (from command area)</i>	299
<i> specifying the quality of m-flagged segments (from command area)</i>	299
<i> EQFITMspecifying the quality of m-flagged segments</i>	299
<i> specifying the quality of m-flagged segments</i>	299
<i> commandsEQFITMspecifying the quality of m-flagged segments</i>	299
<i> EQFITMspecifying the quality of m-flagged segments</i>	299
<i> specifying the quality of m-flagged segments</i>	299
<i> Translation Memorychanging markup, target language and date of segments</i>	301
<i> changing markup, target language and date of segments</i>	301
<i> segmentschanging markup, target language and date</i>	301
<i> changing markup, target language and date</i>	301
<i> markup languagechanging</i>	301
<i> changing</i>	301
<i> EQFTMT</i>	301
<i> commandsEQFTMT</i>	301
<i> EQFTMT</i>	301
<i> EQFTMCL</i>	302
<i> commandsEQFTMCL</i>	302
<i> EQFTMCL</i>	302
<i> EQFREMOVETAGS</i>	303
<i> commandsEQFREMOVETAGS</i>	303
<i> EQFREMOVETAGS</i>	303
<i> SHOWFXP</i>	304
<i> commandsSHOWFXP</i>	304
<i> SHOWFXP</i>	304
<i> EQFINST</i>	304
<i> commandsEQFINST</i>	304
<i> EQFINST</i>	304
<i> CHKCALC</i>	304
<i> commandsCHKCALC</i>	304
<i> CHKCALC</i>	304
<i> EQFADL</i>	305
<i> commandsEQFADL</i>	305
<i> EQFADL</i>	305
<i> EXP2TMX</i>	306
<i> commandsEXP2TMX</i>	306
<i> EXP2TMX</i>	306
<i> TMX2EXP</i>	306
<i> TMX2EXP</i>	306
<i> commandsTMX2EXP</i>	306
<i> commandsTMX2EXP</i>	306
<i> TMX2EXP</i>	306
<i> TMX2EXP</i>	306
<i> ChTypeExportFolder</i>	307
<i> commandsChTypeExportFolder</i>	307
<i> ChTypeExportFolder</i>	307
<i> EQFCLEAR</i>	309
<i> programming interface callsEQFCLEAR</i>	309
<i> EQFCLEAR</i>	309
<i> EQFCLOSE</i>	310
<i> programming interface callsEQFCLOSE</i>	310
<i> EQFCLOSE</i>	310
<i> EQFCONVERTFILENAMES</i>	310
<i> programming interface callsEQFCONVERTFILENAMES</i>	310
<i> EQFCONVERTFILENAMES</i>	310
<i> EQFDELSEG</i>	311
<i> programming interface callsEQFDELSEG</i>	311
<i> EQFDELSEG</i>	311
<i> EQFDICTLOOK</i>	311

<i>programming interface calls</i>	<i>EQFDICTLOOK</i>	311
<i>EQFDICTLOOK</i>		311
<i>EQFILECONVERSIONEX</i>		312
<i>programming interface calls</i>	<i>EQFILECONVERSIONEX</i>	312
<i>EQFILECONVERSIONEX</i>		312
<i>EQGETDICT</i>		313
<i>programming interface calls</i>	<i>EQGETDICT</i>	313
<i>EQGETDICT</i>		313
<i>EQGETDOCFORMAT</i>		314
<i>programming interface calls</i>	<i>EQGETDOCFORMAT</i>	314
<i>EQGETDOCFORMAT</i>		314
<i>EQGETPROP</i>		314
<i>programming interface calls</i>	<i>EQGETPROP</i>	314
<i>EQGETPROP</i>		314
<i>EQGETSENUM</i>		315
<i>programming interface calls</i>	<i>EQGETSENUM</i>	315
<i>EQGETSENUM</i>		315
<i>EQGETSOURCELANG</i>		315
<i>programming interface calls</i>	<i>EQGETSOURCELANG</i>	315
<i>EQGETSOURCELANG</i>		315
<i>EQGETTARGETLANG</i>		315
<i>programming interface calls</i>	<i>EQGETTARGETLANG</i>	315
<i>EQGETTARGETLANG</i>		315
<i>EQFINIT</i>		316
<i>programming interface calls</i>	<i>EQFINIT</i>	316
<i>EQINIT</i>		316
<i>EQQUERYEXITINFO</i>		316
<i>programming interface calls</i>	<i>EQQUERYEXITINFO</i>	316
<i>EQQUERYEXITINFO</i>		316
<i>EQSAVESEG</i>		317
<i>programming interface calls</i>	<i>EQSAVESEG</i>	317
<i>EQSAVESEG</i>		317
<i>EQSEGFILECONVERTASCII2- UNICODE</i>		318
<i>programming interface calls</i>	<i>EQSEGFILECONVERTASCII2- UNICODE</i>	318
<i>EQSEGFILECONVERTASCII2- UNICODE</i>		318
<i>EQSEGFILECONVERTUNICODE2ASCII</i>		319
<i>programming interface calls</i>	<i>EQSEGFILECONVERTUNICODE2ASCII</i>	319
<i>EQSEGFILECONVERTUNICODE2ASCII</i>		319
<i>EQTRANSSEG</i>		319
<i>programming interface calls</i>	<i>EQTRANSSEG</i>	319
<i>EQTRANSSEG</i>		319
<i>EQWORDCNPTRSEG</i>		320
<i>programming interface calls</i>	<i>EQWORDCNPTRSEG</i>	320
<i>EQWORDCNPTRSEG</i>		320
<i>EQWRITEHISTLOG</i>		321
<i>programming interface calls</i>	<i>EQWRITEHISTLOG</i>	321
<i>EQWRITEHISTLOG</i>		321
<i>folderdefinition in non-DDE API</i>		322
<i>definition in non-DDE API</i>		322
<i>subfolderdefinition in non-DDE API</i>		322
<i>definition in non-DDE API</i>		322
<i>EqfAnalyzeDoc</i>		324
<i>programming interface calls</i>	<i>EqfAnalyzeDoc</i>	324
<i>EqfAnalyzeDoc</i>		324
<i>EqfAnalyzeDocEx</i>		326
<i>programming interface calls</i>	<i>EqfAnalyzeDocEx</i>	326
<i>EqfAnalyzeDocEx</i>		326
<i>EqfArchiveTM</i>		327
<i>programming interface calls</i>	<i>EqfArchiveTM</i>	327
<i>EqfArchive TM</i>		327
<i>EqfBuildSegDocName</i>		328
<i>programming interface calls</i>	<i>EqfBuildSegDocName</i>	328
<i>EqfBuildSegDocName</i>		328
<i>EqfChangeFolProps</i>		329
<i>programming interface calls</i>	<i>EqfChangeFolProps</i>	329
<i>EqfChangeFolProps</i>		329
<i>EqfChangeFolPropsEx</i>		330
<i>programming interface calls</i>	<i>EqfChangeFolPropsEx</i>	330
<i>EqfChangeFolPropsEx</i>		330
<i>EqfChangeMFlag</i>		332
<i>programming interface calls</i>	<i>EqfChangeMFlag</i>	332
<i>EqfChangeMFlag</i>		332
<i>EqfCleanMemory</i>		332
<i>programming interface calls</i>	<i>EqfCleanMemory</i>	332
<i>EqfCleanMemory</i>		332
<i>EqfCountWords</i>		334
<i>programming interface calls</i>	<i>EqfCountWords</i>	334
<i>EqfCountWords</i>		334

<i>EqfCreateCntReport</i>	335
<i>programming interface callsEqfCreateCntReport</i>	335
<i>EqfCreateCntReport</i>	335
<i>EqfCreateCountReport</i>	339
<i>programming interface callsEqfCreateCountReport</i>	339
<i>EqfCreateCountReport</i>	339
<i>EqfCreateControlledFolder</i>	340
<i>programming interface callsEqfCreateControlledFolder</i>	340
<i>EqfCreateControlledFolder</i>	340
<i>EqfCreateFolder</i>	342
<i>programming interface callsEqfCreateFolder</i>	342
<i>EqfCreateFolder</i>	342
<i>EqfCreateITM</i>	343
<i>programming interface callsEqfCreateITM</i>	343
<i>EqfCreateITM</i>	343
<i>EqfCreateMem</i>	346
<i>programming interface callsEqfCreateMem</i>	346
<i>EqfCreateMem</i>	346
<i>EqfCreateSubFolder</i>	347
<i>programming interface callsEqfCreateSubFolder</i>	347
<i>EqfCreateSubFolder</i>	347
<i>EqfDeleteDoc</i>	348
<i>programming interface callsEqfDeleteDoc</i>	348
<i>EqfDeleteDoc</i>	348
<i>EqfDeleteFolder</i>	349
<i>programming interface callsEqfDeleteFolder</i>	349
<i>EqfDeleteFolder</i>	349
<i>EqfDeleteMem</i>	349
<i>programming interface callsEqfDeleteMem</i>	349
<i>EqfDeleteMem</i>	349
<i>EqfEndSession</i>	350
<i>programming interface callsEqfEndSession</i>	350
<i>EqfEndSession</i>	350
<i>EqfExportDict</i>	350
<i>programming interface callsEqfExportDict</i>	350
<i>EqfExportDict</i>	350
<i>EqfExportDoc</i>	351
<i>programming interface callsEqfExportDoc</i>	351
<i>EqfExportDoc</i>	351
<i>EqfExportFolder</i>	352
<i>programming interface callsEqfExportFolder</i>	352
<i>EqfExportFolder</i>	352
<i>EqfExportFolderFP</i>	353
<i>programming interface callsEqfExportFolderFP</i>	353
<i>EqfExportFolderFP</i>	353
<i>EqfExportFolderFPAs</i>	355
<i>programming interface callsEqfExportFolderFPAs</i>	355
<i>EqfExportFolderFPAs</i>	355
<i>EqfExportMem</i>	356
<i>programming interface callsEqfExportMem</i>	356
<i>EqfExportMem</i>	356
<i>EqfFreeSegFile</i>	357
<i>programming interface callsEqfFreeSegFile</i>	357
<i>EqfFreeSegFile</i>	357
<i>EqfGetFolderProp</i>	358
<i>programming interface callsEqfGetFolderProp</i>	358
<i>EqfGetFolderProp</i>	358
<i>EqfGetLastError</i>	359
<i>programming interface callsEqfGetLastError</i>	359
<i>EqfGetLastError</i>	359
<i>EqfGetMatchLevel</i>	359
<i>programming interface callsEqfGetMatchLevel</i>	359
<i>EqfGetMatchLevel</i>	359
<i>EqfGetProgress</i>	361
<i>programming interface callsEqfGetProgress</i>	361
<i>EqfGetProgress</i>	361
<i>EqfGetSegNum</i>	362
<i>programming interface callsEqfGetSegNum</i>	362
<i>EqfGetSegNum</i>	362
<i>EqfGetSegW</i>	363
<i>programming interface callsEqfGetSegW</i>	363
<i>EqfGetSegW</i>	363
<i>EqfGetSegmentNumber</i>	364
<i>programming interface callsEqfGetSegmentNumber</i>	364
<i>EqfGetSegmentNumber</i>	364
<i>EqfGetSourceLine</i>	364
<i>programming interface callsEqfGetSourceLine</i>	364
<i>EqfGetSourceLine</i>	364

<i>EqfGetSysLanguage</i>	365
<i>programming interface callsEqfGetSysLanguage</i>	365
<i>EqfGetSysLanguage</i>	365
<i>EqfGetVersion</i>	366
<i>programming interface callsEqfGetVersion</i>	366
<i>EqfGetVersion</i>	366
<i>EqfImportDoc</i>	366
<i>programming interface callsEqfImportDoc</i>	366
<i>EqfImportDoc</i>	366
<i>EqfImportDict</i>	368
<i>programming interface callsEqfImportDict</i>	368
<i>EqfImportDict</i>	368
<i>EqfImportFolder</i>	369
<i>programming interface callsEqfImportFolder</i>	369
<i>EqfImportFolder</i>	369
<i>EqfImportFolderFP</i>	370
<i>programming interface callsEqfImportFolderFP</i>	370
<i>EqfImportFolderFP</i>	370
<i>EqfImportMem</i>	371
<i>EqfImportMem</i>	371
<i>programming interface callsEqfImportMem</i>	371
<i>EqfImportMem</i>	371
<i>EqfImportMem</i>	371
<i>EqfImportMem</i>	371
<i>EqfLoadSegFile</i>	372
<i>programming interface callsEqfLoadSegFile</i>	372
<i>EqfLoadSegFile</i>	372
<i>EqfOrganizeMem</i>	373
<i>programming interface callsEqfOrganizeMem</i>	373
<i>EqfOrganizeMem</i>	373
<i>EqfProcessNomatch</i>	373
<i>programming interface callsEqfProcessNomatch</i>	373
<i>EqfProcessNomatch</i>	373
<i>EqfRename</i>	375
<i>programming interface callsEqfRename</i>	375
<i>EqfRename</i>	375
<i>EqfSetSysLanguage</i>	376
<i>programming interface callsEqfSetSysLanguage</i>	376
<i>EqfSetSysLanguage</i>	376
<i>EqfStartSession</i>	377
<i>programming interface callsEqfStartSession</i>	377
<i>EqfStartSession</i>	377
<i>EqfUpdateSegW</i>	377
<i>programming interface callsEqfUpdateSegW</i>	377
<i>EqfUpdateSegW</i>	377
<i>EqfWriteSegFile</i>	378
<i>programming interface callsEqfWriteSegFile</i>	378
<i>EqfWriteSegFile</i>	378
<i>creatingtables of contents</i>	383
<i>tables of contents</i>	383
<i>table of contentscreating</i>	383
<i>creating</i>	383
<i>documentcreating its table of contents</i>	383
<i>creating its table of contents</i>	383
<i>user exitentry pointEQFPRESEGW</i>	386
<i>entry pointEQFPRESEGW</i>	386
<i>EQFPRESEGW</i>	386
<i>user exitentry pointEQFPOSTSEGW</i>	386
<i>entry pointEQFPOSTSEGW</i>	386
<i>EQFPOSTSEGW</i>	386
<i>user exitentry pointEQFPOSTTMW</i>	386
<i>entry pointEQFPOSTTMW</i>	386
<i>EQFPOSTTMW</i>	386
<i>user exitentry pointEQFCHECKSEGW</i>	386
<i>entry pointEQFCHECKSEGW</i>	386
<i>EQFCHECKSEGW</i>	386
<i>user exitentry pointEQFSHOW</i>	386
<i>entry pointEQFSHOW</i>	386
<i>EQFSHOW</i>	386
<i>user exitentry pointEQFPREUNSEGW</i>	386
<i>entry pointEQFPREUNSEGW</i>	386
<i>EQFPREUNSEGW</i>	386
<i>user exitentry pointEQFPOSTUNSEG2</i>	386
<i>entry pointEQFPOSTUNSEG2</i>	386
<i>EQFPOSTUNSEG2</i>	386
<i>programming interface callscompatibility notes</i>	386
<i>compatibility notes</i>	386
<i>programming interface callsEQFPRESEG2</i>	386

<i>EQFPRESEG2</i>	386
<i>EQFPRESEG2</i>	386
<i>programming interface calls</i> <i>EQFPRESEGEX</i>	387
<i>EQFPRESEGEX</i>	387
<i>EQFPRESEGEX</i>	387
<i>programming interface calls</i> <i>EQFPOSTSEGW</i>	388
<i>EQFPOSTSEGW</i>	388
<i>EQFPOSTSEGW</i>	388
<i>programming interface calls</i> <i>EQFPOSTSEGWEX</i>	388
<i>EQFPOSTSEGWEX</i>	388
<i>EQFPOSTSEGWEX</i>	388
<i>EQFPOSTSEGWEX</i>	388
<i>programming interface calls</i> <i>EQFPOSTTMW</i>	389
<i>EQFPOSTTMW</i>	389
<i>EQFPOSTTMW</i>	389
<i>programming interface calls</i> <i>EQFCHECKSEGW</i>	389
<i>EQFCHECKSEGW</i>	389
<i>EQFCHECKSEGW</i>	389
<i>programming interface calls</i> <i>EQFSHOW</i>	390
<i>EQFSHOW</i>	390
<i>EQFSHOW</i>	390
<i>programming interface calls</i> <i>EQFGETCURSEG</i>	390
<i>EQFGETCURSEG</i>	390
<i>EQFGETCURSEG</i>	390
<i>programming interface calls</i> <i>EQFGETCURSEGW</i>	390
<i>EQFGETCURSEGW</i>	390
<i>EQFGETCURSEGW</i>	390
<i>programming interface calls</i> <i>EQFGETNEXTSEG</i>	391
<i>EQFGETNEXTSEG</i>	391
<i>EQFGETNEXTSEG</i>	391
<i>programming interface calls</i> <i>EQFGETNEXTSEGW</i>	391
<i>EQFGETNEXTSEGW</i>	391
<i>EQFGETNEXTSEGW</i>	391
<i>programming interface calls</i> <i>EQFGETPREVSEG</i>	392
<i>EQFGETPREVSEG</i>	392
<i>EQFGETPREVSEG</i>	392
<i>programming interface calls</i> <i>EQFGETPREVSEGW</i>	392
<i>EQFGETPREVSEGW</i>	392
<i>EQFGETPREVSEGW</i>	392
<i>programming interface calls</i> <i>EQFBUILDDOCPATH</i>	392
<i>EQFBUILDDOCPATH</i>	392
<i>EQFBUILDDOCPATH</i>	392
<i>programming interface calls</i> <i>EQFGETINFO</i>	393
<i>EQFGETINFO</i>	393
<i>EQFGETINFO</i>	393
<i>programming interface calls</i> <i>EQFPREUNSEGW</i>	394
<i>EQFPREUNSEGW</i>	394
<i>EQFPREUNSEGW</i>	394
<i>programming interface calls</i> <i>EQFPOSTUNSEGW</i>	395
<i>EQFPOSTUNSEGW</i>	395
<i>EQFPOSTUNSEGW</i>	395
<i>programming interface calls</i> <i>EQFPOSTUNSEG2</i>	395
<i>EQFPOSTUNSEG2</i>	395
<i>EQFPOSTUNSEG2</i>	395
<i>API calls for user exits</i> <i>API calls for user exits</i>	396
<i>API calls for user exits</i>	396
<i>programming interface calls</i> <i>EQFGETTAOPTIONS</i>	396
<i>EQFGETTAOPTIONS</i>	396
<i>EQFGETTAOPTIONS</i>	396
<i>programming interface calls</i> <i>EQFSETTAOPTIONS</i>	396
<i>EQFSETTAOPTIONS</i>	396
<i>EQFSETTAOPTIONS</i>	396
<i>programming interface calls</i> <i>EQFTAOPTIONS</i>	396
<i>EQFTAOPTIONS</i>	396
<i>EQFTAOPTIONS</i>	396
<i>Lotus Notes markup table</i> <i>with user exit</i>	397
<i>markup table with user exit</i>	397
<i>markup table with user exit</i>	397
<i>user exit for</i> <i>LOTUSNGD markup table</i>	397
<i>for LOTUSNGD markup table</i>	397
<i>LOTUSNGD markup table</i> <i>user exit for</i>	397
<i>user exit for</i>	397
<i>context information in Lotus Notes elements</i>	397
<i>in Lotus Notes elements</i>	397
<i>user exit entry point</i> <i>EQFGETCONTEXTINFO</i>	397
<i>entry point</i> <i>EQFGETCONTEXTINFO</i>	397
<i>user exit entry point</i> <i>EQFGETSEGCONTEXT</i>	397
<i>entry point</i> <i>EQFGETSEGCONTEXT</i>	397

EQFGETSEGCONTEXT.....	397
user exit entry pointEQFUPDATECONTEXT	397
entry pointEQFUPDATECONTEXT	397
EQFUPDATECONTEXT.....	397
user exit entry pointEQFCOMPARECONTEXT	397
entry pointEQFCOMPARECONTEXT	397
EQFCOMPARECONTEXT.....	397
programming interface callsEQFGETCONTEXTINFO	397
EQFGETCONTEXTINFO.....	397
programming interface callsEQFGETSEGCONTEXT	397
EQFGETSEGCONTEXT.....	397
EQFGETSEGCONTEXT.....	397
programming interface callsEQFUPDATECONTEXT	398
EQFUPDATECONTEXT.....	398
EQFUPDATECONTEXT.....	398
programming interface callsEQFCOMPARECONTEXT	398
EQFCOMPARECONTEXT.....	398
EQFCOMPARECONTEXT.....	398
Unicodesupport for parser API	399
support for parser API.....	399
parser APIUnicode support	399
Unicode support.....	399
programming interface callsParsInitialize	399
ParsInitialize.....	399
ParsInitialize.....	399
parser APIParsInitialize	399
ParsInitialize.....	399
programming interface callsParsBuildTempName	399
ParsBuildTempName.....	399
ParsBuildTempName.....	399
parser APIParsBuildTempName	399
ParsBuildTempName.....	399
programming interface callsParsLoadSegFile	400
ParsLoadSegFile.....	400
ParsLoadSegFile.....	400
parser APIParsLoadSegFile	400
ParsLoadSegFile.....	400
programming interface callsParsGetSegNum	400
ParsGetSegNum.....	400
ParsGetSegNum.....	400
parser APIParsGetSegNum	400
ParsGetSegNum.....	400
programming interface callsParsGetSeg	400
ParsGetSeg.....	400
ParsGetSeg.....	400
parser APIParsGetSeg	400
ParsGetSeg.....	400
programming interface callsParsGetSegW	401
ParsGetSegW.....	401
ParsGetSegW.....	401
parser APIParsGetSegW	401
ParsGetSegW.....	401
programming interface callsParsUpdateSeg	401
ParsUpdateSeg.....	401
ParsUpdateSeg.....	401
parser APIParsUpdateSeg	401
ParsUpdateSeg.....	401
programming interface callsParsUpdateSegW	401
ParsUpdateSegW.....	401
ParsUpdateSegW.....	401
parser APIParsUpdateSegW	401
ParsUpdateSegW.....	401
programming interface callsParsWriteSegFile	402
ParsWriteSegFile.....	402
ParsWriteSegFile.....	402
parser APIParsWriteSegFile	402
ParsWriteSegFile.....	402
programming interface callsParsMakeSGMLSegment	402
ParsMakeSGMLSegment.....	402
ParsMakeSGMLSegment.....	402
parser APIParsMakeSGMLSegment	402
ParsMakeSGMLSegment.....	402
programming interface callsParsMakeSGMLSegmentW	403
ParsMakeSGMLSegmentW.....	403
ParsMakeSGMLSegmentW.....	403
parser APIParsMakeSGMLSegmentW	403
ParsMakeSGMLSegmentW.....	403

<i>programming interface callsParsConvert</i>	403
<i>ParsConvert</i>	403
<i>ParsConvert</i>	403
<i>parser APIParsConvert</i>	403
<i>ParsConvert</i>	403
<i>programming interface callsParsGetDocName</i>	404
<i>ParsGetDocName</i>	404
<i>ParsGetDocName</i>	404
<i>parser APIParsGetDocName</i>	404
<i>ParsGetDocName</i>	404
<i>programming interface callsParsGetDocLang</i>	404
<i>ParsGetDocLang</i>	404
<i>ParsGetDocLang</i>	404
<i>parser APIParsGetDocLang</i>	404
<i>ParsGetDocLang</i>	404
<i>programming interface callsParsSplitSeg</i>	404
<i>ParsSplitSeg</i>	404
<i>ParsSplitSeg</i>	404
<i>parser APIParsSplitSeg</i>	404
<i>ParsSplitSeg</i>	404
<i>programming interface callsParsSplitSegW</i>	405
<i>ParsSplitSegW</i>	405
<i>ParsSplitSegW</i>	405
<i>parser APIParsSplitSegW</i>	405
<i>ParsSplitSegW</i>	405
<i>programming interface callsParsFreeSegFile</i>	405
<i>ParsFreeSegFile</i>	405
<i>ParsFreeSegFile</i>	405
<i>parser APIParsFreeSegFile</i>	405
<i>ParsFreeSegFile</i>	405
<i>programming interface callsParsLoadMarkup</i>	406
<i>ParsLoadMarkup</i>	406
<i>ParsLoadMarkup</i>	406
<i>parser APIParsLoadMarkup</i>	406
<i>ParsLoadMarkup</i>	406
<i>programming interface callsPars_tokenize</i>	406
<i>Pars_tokenize</i>	406
<i>Pars_tokenize</i>	406
<i>parser APIPars_tokenize</i>	406
<i>Pars_tokenize</i>	406
<i>programming interface callsPars_tokenizeW</i>	406
<i>Pars_tokenizeW</i>	406
<i>Pars_tokenizeW</i>	406
<i>parser APIPars_tokenizeW</i>	406
<i>Pars_tokenizeW</i>	406
<i>programming interface callsParsGetNextToken</i>	407
<i>ParsGetNextToken</i>	407
<i>ParsGetNextToken</i>	407
<i>parser APIParsGetNextToken</i>	407
<i>ParsGetNextToken</i>	407
<i>programming interface callsParsFreeMarkup</i>	408
<i>ParsFreeMarkup</i>	408
<i>ParsFreeMarkup</i>	408
<i>parser APIParsFreeMarkup</i>	408
<i>ParsFreeMarkup</i>	408
<i>programming interface callsParsTerminate</i>	408
<i>ParsTerminate</i>	408
<i>ParsTerminate</i>	408
<i>parser APIParsTerminate</i>	408
<i>ParsTerminate</i>	408
<i>font family</i> for automatic font substitution	426
<i>for automatic font substitution</i>	426
<i>bidirectional languages</i> limitations help compiler	426
<i>limitations</i> help compiler	426
<i>help compiler</i>	426
<i>help compiler</i>	426
<i>limitations</i> help compiler	426
<i>help compiler</i>	426