RC-WinTrans 9

Professional Software Localization Tool
for Windows Win32, Microsoft .NET,
Java, XML and custom file types

– User Manual –

January 2010

Contents

Chapter 1

Introduction

1 Introduction

1.1 RC-WinTrans9

RC-WinTrans 9 is a localization tool for software created with Windows Win32, Microsoft .NET, and Java software development platforms. It provides a highly specialized environment for localizing (creating, editing, and maintaining) an application's graphical user interface (GUI) into other languages. RC-WinTrans features easy navigation plus everything needed to translate/localize text, menus, dialog boxes and .NET Forms, accelerator keys, VersionInfo (Win32), bitmap and message table resources (Win32).

Originally designed in 1995 to translate RC files, RC-WinTrans has grown to support Java properties files, Microsoft and Borland .NET files, Win32 binary executable (EXE/DLL) files, Windows INI and InstallShield SHL files, and XML* files as well as a number of others (*with restrictions). For over 15 years prominent companies across the globe have trusted RC-WinTrans to produce high-quality, localized versions of their software. During this time we have gained valuable experience with the different issues related to the localization process and with the methods used to resolve them. It is this experience which has gone into making each successive and improved version of RC-WinTrans a reliable choice for all types of software localization projects.

The latest version of RC-WinTrans is RC-WinTrans 9, our most powerful localization solution to date. It features an advanced system design with automization, extension, and customization options plus a full set of translation and project management functions for the file types listed below.

Supported File Types

.NET All types of .NET XML (RESX) and binary resource files (EXE, DLL, RESOURCES) for .NET 1.1 and 2.0; text only in .NET 3.0 and 3.5. Also supported: .NET files from **Borland Delphi**.

Win32 Windows resource script (RC) and binary program/library files (EXE/DLL/OCX); ActiveX controls.

INI, SHL Windows INI files, InstallShield stringtable files, and Unix INI files.

64-bit (x64) application files including Windows (EXE/DLL) and .NET Windows Form s (EXE/DLL).

Java Java properties files.

XML XML and XSL files supported with XSLT transformation files.

(Continued on next page.)

Supported File Types (Continued from previous page.)

C/C++ header files.

WPF BAML (Windows Presentation Foundation files) Text only- no GUI translation support.

ADM Windows ADM (Administrative Template) files.

PO GNU PO format files that are textual and modifiable.

CSV Tab-separated text files (format: <name/ID> TAB <text>).

Other Other file types supported by custom converters. Contact Schaudin.com for information and/or a free analysis.

Supported file types are discussed in greater detail in Section 3.1 of this guide. Information on creating a translation project in order to localize resource files can be found in Section 3.2.2, "Creating a Translation Project."

Views

RC-WinTrans is a multiple-document interface (MDI) application that lets you open and work with several translation projects at the same time. RC-WinTrans' main window (the *Project* window) presents the text to be translated in an coherent table display (the *Text Table* view) with a column for the source text, the target text, and the item ID, as well as for any additional information which may be available. Special WYSIWYG (**W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et) views for menu and dialog box resources and .NET Forms allow these elements to be displayed and translated in their proper contexts. Other views show repetitions (multiple, identical source texts), translations for each project language, all the elements contained in a group, source data, and statistical information.

Additional details can be found in Section 3.3.1, "The *Project* Window."

***Translation Edit* Bar**

The *Translation Edit* bar is the primary place for entering and editing translation text. All of RC-WinTrans' resource views (*Text Table*, *Menu*, and *Dialog Box*) use the *Translation Edit* bar to present and translate a selected text resource (in addition to the ability to enter/edit translations within the views themselves). Other resources such as stringtable texts are always presented in the same way and in the same place (*Text Table* view/*Translation Edit* bar).

Not only does the *Translation Edit* bar support the translation of text resources, it also handles Version info entries, accelerator keys, and image resources. For more information see Section 3.3.2, "*Translation Edit* Bar."

.NET Windows Forms

RC-WinTrans supports the localization of Microsoft .NET Windows Forms applications as well as control library and resource files developed in C# or Visual Basic for .NET. Windows Forms can be displayed and edited via RC-WinTrans' Forms Viewer and External .NET Forms Editor. A specialized view for menus is also available. Versions 1.1 and 2.0 of the .NET Framework are fully supported while Versions 3.0 and 3.5 are supported for text translation only.

Please refer to Section 3.3.4, "*Dialog Box* View and Dialog Box Editor," and Section 3.3.5, ".NET Forms Editor," for further details.

Dialog Box Editor (Win32)/.NET Forms Editor

RC-WinTrans comes with a built-in Dialog Box and Forms Editor that lets you modify the layout of dialog boxes and their individual elements to fit the requirements of the translation text. The user will be informed of size adjustments. The font and style of a dialog box can also be changed to suit the translation text.

Additional information can be found in Section 3.3.4, "*Dialog Box* View and Dialog Box Editor," and in Section 3.3.5, ".NET Forms Editor."

ActiveX Controls (Win32)

RC-WinTrans is the only localization tool that supports the translation of ActiveX control properties. An advanced mechanism extracts all of the text associated with an ActiveX control and allows it to be translated and/or edited with RC-WinTrans. Once the text has been extracted it can be translated without having to install the ActiveX controls on the translator's computer, thus eliminating the need to distribute and install the controls to/for everyone involved in the project.

For a more in-depth description of ActiveX control handling see the RC-WinTrans Help topic "Localizing ActiveX Control Data."

Dictionaries

Translators can use integrated RC-WinTrans resource text dictionaries as well as their own dictionaries (in TXT or TMX format) to assist them during the translation process. The *Automatic Translation* and *Import from Dictionary* commands enable automatic translation based on the data of one or more dictionaries. You can also use an existing translation project in the same way as a dictionary, so text doesn't have to be translated twice, and the amount of data available to translate current projects increases with the number of translated projects. See Section 3.6, "Dictionaries," for more information.

Trados Translation Memories

RC-WinTrans' integrated use of the Trados™ Translator's Workbench lets you access a Trados translation memory from within RC-WinTrans to find, get, and save translations.

For further details see Section 3.7, "Trados and RC-WinTrans."

Automatic Translation

Use selected data sources (dictionaries/translation memories and/or open translation projects) to automatically translate text items. Minimum match values can be determined and the data sources selected in *the Dictionary Options* dialog box. You can even assign a specific state (e.g., "needs review translation") or a marker to items that are translated automatically so that they'll be easy to find and review.

Spell Checker

RC-WinTrans' spell check function supports western languages with ready-to-use lexicons for Dutch, Danish, German, French, Finnish, Portuguese, Brazilian, Italian, English UK, English US, Norwegian Nynorsk, Swedish, and Spanish. Alternatively, the spell check function from Microsoft Word can be used (see Section 3.5.1.1, "Spell Check").

Check Functions

RC-WinTrans is equipped with a set of check functions to support high-quality software localization that is both time-efficient and cost-effective. Additional information can be found in Section 3.5.1 of this guide ("Check Functions").

Reports and Statistics

RC-WinTrans provides statistics and checks reports created and formatted in HTML. Reports can be created for the current project file (translation database) or for all of the project files in a workspace at once.

More details can be found in Section 3.5.2, "Reports and Statistics."

Import and Export/Distributed Translation

A set of import and export functions for distributing and merging translation data is also included. Direct import and export for text and TMX files, Microsoft Excel spreadsheets, and RC-WinTrans translation databases are supported. These are particularly useful in situations where all or part of a project needs to be given to various team members then reintegrated into the main project before the final version of the localized software can be created.

Additional information is available in Section 3.8, "Import and Export," and in Chapter 4, "External Translation."

Changes Control

One of the main advantages to using RC-WinTrans is the changes control feature which tracks changes in the source file and points out any instances in which they affect existing translation(s). Changes and newly added items are made visible to the translator in several different ways so that they can easily be found and reviewed/translated.

Please refer to Section 3.4, "Source File Changes Handling," for details.

Project Management

Files can be organized and dealt with more effectively by grouping them together in projects and workspaces. One or more files can be contained in a project and, accordingly, one or more projects can be contained in a workspace.

See Section 3.2, "Translation Projects," for more information.

Visual Basic For Applications (VBA)

The RC-WinTrans localization environment has been expanded to include Microsoft Visual Basic for Applications (VBA). VBA technology in combination with RC-WinTrans' COM object model (Section 5.3) makes RC-WinTrans extendable, customizable, and capable of being adapted to suit virtually any type of localization project.

For additional details see Section 5.1, "Microsoft Visual Basic For Applications."

Add-Ons for Special/Extended Functionality

RC-WinTrans Add-Ons (Section 5.3) extend RC-WinTrans' capabilities to carry out specialized tasks and functions not already included in the application. Add-ons have their own menu in RC-WinTrans to access the specified functionality. The integration is seamless and can be realized with a minimum of time, cost and effort thanks to RC-WinTrans' advanced technological design. Please contact Schaudin.com for more information (contact and support: Chapter 7).

Command Line Support

RC-WinTrans offers the following two options for automating processes via batch processing:

1. *Command Line Tool*
   RC-WinTrans comes with a command line tool “RWTCmd.exe” which provides a set of command line commands that can be called from a batch process or from the command prompt. The command line tool is a small program which does not execute commands itself. Instead it uses RC-WinTrans as an automation server and RC-WinTrans executes the command.

   The available command line commands are described in the RC-WinTrans Help (“Command Line Commands” topic).

2. *Windows Scripting*
   Script files in any language supported by the Windows Scripting Host can be used for scripting. Examples of such scripting languages include VBScript, Jscript, Perl Script, etc. A script can make use of RC-WinTrans’ extended COM automation interface which lets processes be automated in a wide variety of ways.

Please refer to Chapter 6, “Command Line Processing,” for further details.

1.2 System Requirements

1.2.1 Minimum Hardware

- CPU: 1800 MHz
- Memory: 256 MB RAM (recommended: 512 MB – 1GB)
- Display: 1024 x 768, high color (16 bit)
- Monitor: 17” (recommended: 19”)

1.2.2 Operating System

RC-WinTrans is a 32-bit Unicode application and can only be installed on a PC running on one of the following operating systems:

- Windows 2000 with service pack 4 (SP4)
  Windows Server 2003
- Windows XP
- Windows Vista
- Windows 7

Windows operating systems Windows 95, Windows 98, and Windows Me are not supported.

1.2.3 Software

- Microsoft Internet Explorer 6.0 (or higher)- required for the RC-WinTrans graphical user interface.

NOTE: The installation program of RC-WinTrans will prompt a message if Internet Explorer 6.0 is not installed on the target system. The installation will be cancelled if Internet Explorer 6.0 is not available.

- Adobe SVG Viewer (for the graphical portions of the HTML statistics views)

NOTE: The SVG Viewer can be downloaded for free from Adobe’s Website. An installation file for the SVG Viewer 3.1 is included with the RC-WinTrans installation CD.

- Microsoft .NET Framework 1.1 or .NET Framework 2.0 (required run-time environment)

- Microsoft .NET Framework 2.0

The .NET Framework 2.0 is required to process the .NET assembly files (EXE/DLL) that were created for it with Visual Studio 2005. The .NET Framework 2.0 installation is included in the current installation package of RC-WinTrans.

1.2.4 Optional Software

- Microsoft Excel (data import/export)
- Microsoft Word (spell check and other text options- 2002 version or later)
- Trados* Translator's Workbench (Versions 3.0 through 7.5)
  * Now SDL TRADOS: www.sdl.com.

1.3 RC-WinTrans Licenses and Editions

RC-WinTrans 9 comes in the following editions:

RC-WinTrans 9 Global Edition

The Global Edition is the complete RC-WinTrans localization environment intended for use by a single user at a designated workstation. All of RC-WinTrans' translation, quality assurance and project management features are available for localizing any number of software files, quickly, easily, and efficiently. This edition also includes the Microsoft VBA integrated development environment.

RC-WinTrans 9 Translator Edition

The Translator Edition offers the same localization environment for RC-WinTrans translation projects that have first been created with the Global Edition, making it ideal for use by transators with whom data must be exchanged. It has only two restrictions on use: a new translation project cannot be created with the Translator Edition, nor is it possible to add or remove files from an existing project. The Translator Edition is also a single-user license intended for use at a designated workstation.

Unlike other software localization tools, RC-WinTrans does not use specially-prepared, proprietary "bundle" files for translators. The Translator Edition works with the same project and project data files as the Global Edition. Project data files can be packed into a standard format ZIP file to be sent to external translators, then returned in the same manner after being opened, edited and saved.

Translator Group License

The Translator Group License designed to accommodate companies that use a number of internal or external translators in their business. It supplies one single-user license of the Global Edition to address project management needs and one multi-user license of the Translator Edition to be distributed to as many different translators as is required.

The Translator Group License is an alternative to purchasing multiple copies of the RC-WinTrans Global Edition. A company may purchase additional copies of the single-user-licensed Global Edition for users who require full functionality at a reduced price.

The Translator Group License is offered as an alternative to purchasing multiple copies of the Version 9 Global Edition. A company may purchase additional copies of the single-user-licensed Global Edition for users who require full functionality. We offer additional Global Edition licenses at a comparatively lower price for holders of the Translator Group License.

1.4 Materials

RC-WinTrans can be purchased as a download-only software package (no materials included) or as a physical product package containing the following materials:

- CD-ROM
- Product and licensing information.

1.5 Setup

RC-WinTrans 9 comes with a setup program that installs the RC-WinTrans software and the third-party components/programs required by the application. The Microsoft .NET Framework 1.1 is installed as the first package (i.e., the RC-WinTrans setup starts the Microsoft .NET Framework installation).

NOTE: The .NET installation has its own behavior and user interface. Once it has been installed it requires confirmation of a license agreement from Microsoft.

Installation Using the Download File

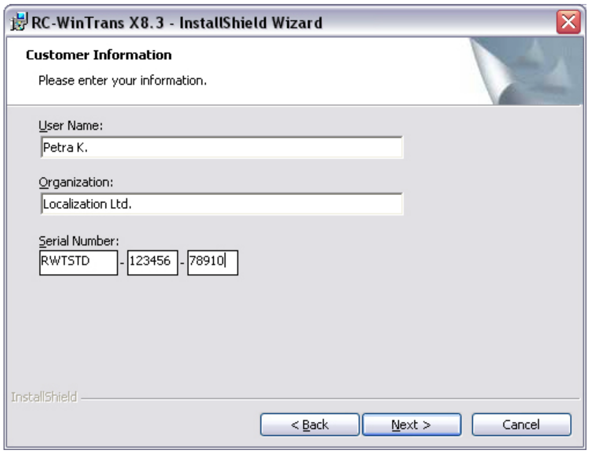
Run the downloaded RC-WinTrans 9 installation file, a self-extracting executable (.exe) file that starts the setup.

Installation from the CD

Inserting the RC-WinTrans 9 installation CD into your computer's CD drive automatically starts the setup and prompts a dialog box with a command to start the installation.

Serial Number (License Key)

During the setup you will be prompted to enter your serial number (license key). You may have received the license key number with your e-mail order confirmation or, if a software package was sent, the key can be found on the inside/backside of the CD case.



RC-WinTrans X8.3 - InstallShield Wizard

Customer Information

Please enter your information.

User Name: Petra K.

Organization: Localization Ltd.

Serial Number: RWTSTD - 123456 - 78910

Figure: The Installation Wizard's *Customer Information* page where the serial number (license key) for the RC-WinTrans 9 product must be entered in order to activate the software.

Product Languages

The RC-WinTrans user interface is available in English and French. The help system and other documentation is only available in English. The setup procedure is also English-only, but you can select either English or French as the language of the installed software. The language can be selected on the *User Interface* tab in the *Options* dialog box. This page can be accessed via the *Options* command under the *Tools* menu.

Notes on Installed Components

1. ***Sample Translation Project***

   A sample translation project is included in the "Samples\Translation\Project1" folder of the RC-WinTrans installation directory. Open the project by starting RC-WinTrans and using the *Open* command from the *File* menu, then select the project file in the sample project called "Project1.rwtproject." The project contains several types of English source files that are partially or completely translated into German or Japanese.

2. ***Microsoft Visual Basic for Applications (VBA)***

   The RC-WinTrans setup includes the installation of the VBA run-time environment and the Visual Basic integrated development environment (IDE). As with the RC-WinTrans application, the VBA environment is available in either English and French.

Chapter 2

Migrating from Previous Versions of RC-WinTrans

2 Migrating from Previous Versions of RC-WinTrans

This chapter discusses the major differences between RC-WinTrans 9 and previous versions as well as some things to consider for the update process.

2.1 Data Migration

2.1.1 Translation Databases

RC-WinTrans 9 (and X8 before it) diverges from previous versions with a different method of storing project data (single database file or multiple XLIFF files). Versions 4 through 7 of RC-WinTrans used a Microsoft Access database (MDB, EDB, TDB, XDB files). RC-WinTrans 9 **cannot** open or edit these previously-used translation databases. The data of a previous RC-WinTrans project database can be added (and thereby reused) in RC-WinTrans 9- this is the process known as “data migration.”

Use the *Data Migration: RC-WinTrans 5 – 7 Database* command in the *File* menu to migrate data from an older version of RC-WinTrans. For more information please refer to the RC-WinTrans Help topic: “Project Data Migration (RC-WinTrans 4 – 7)”).

2.1.2 Dictionaries

The format of RC-WinTrans dictionaries (dictionary data) are now text files (TXT) and TMX files. Previous versions of RC-WinTrans used a Microsoft Access database (DDB) for dictionary data- these dictionaries are no longer supported. The data of the previous dictionaries can be reused by exporting the dictionary data (with the old version of RC-WinTrans) to a text file, thereby creating an RC-WinTrans 9 text file dictionary. The exported text file must then be prepared in order to make it available as a Version 9 text dictionary by adding a header line with the language information (language code) and separating the source and target texts with a tab control character. Enclosing the text data of the source and target texts in quotation marks is optional.

See also the RC-WinTrans Help topic “Text Dictionaries.”

2.2 Command Line Tool

The batch command line interface of RC-WinTrans has undergone some minor changes since previous versions. The most important changes are listed here:

- User-defined batch functions can be added with VBA macros, i.e., the command line interface is extendable.
- Language IDs are now specified as ISO 639 language codes, e.g., “en-us,” or “es-jp” (previous versions would expect a four-digit hexadecimal number).

  Almost all commands use a translation project file name (e.g., “myproject.rwtproject”) as the first parameter.
- There is a parameter (“/KeepAlive”) that allows RC-WinTrans to keep running between two command calls.
- A more intuitive rule for return values has been chosen: a positive value is used to signal a successful call while a negative value indicates an error.
- It is possible to programmatically “kill” a running instance of RC-WinTrans in case of a problem.
- A set of script sample batch files can be found in the RC-WinTrans directory under “Samples/Commandline/Batch.”

Please refer to Chapter 6 of this guide (“Command Line Processing”) and the RC-WinTrans Help topic “Command Line Commands” for more information.

2.3 Trados

The functionality for SDL TRADOS translation memories is still the same as in RC-WinTrans 7.

Section 3.7 of this guide deals with RC-WinTrans 9’s Trados integration.

Chapter 3

Features and Functions

3 Features and Functions

3.1 Supported File Types

3.1.1 Windows Win32 Files

FILE TYPE	DESCRIPTION	TARGET FILE TYPE
RC, DLG	Windows resource script (RC) files	Same as source file. Encoding: ANSI/MBCS, UTF-16.
EXE, DLL, OCX	Program/library files	Same as source file-single-language and muliple-language.
MC	Message text files	Same as source file. Encoding: ANSI/MBCS, UTF-16.
ADM	Administrative template file format	Same as source file. Encoding: ANSI/MBCS, UTF-16.
C, CPP	C/C++ header files with (only) string definitions	Same as source file.

A Win32 program/library file must be based on (created with) Windows resource script files in order to be localized with RC-WinTrans. These are, for example, programs developed with the C/C++ programming language. Strings in Visual Basic programs (up to VB6) can be translated if a resource script file has been used to separate the text from the program's code.

NOTE: RC-WinTrans also supports 64-bit (x64) application files..

3.1.2 Microsoft .NET Files

FILE TYPE	DESCRIPTION	TARGET FILE TYPE
RESX	.NET Windows Form, XML resource file.	Same as source file.
RESOURCES	Binary resource file; compiled RESX (XML) file.	Same as source file.
TXT	.NET text file- text only. File contains the ID, text, and, optionally, a comment for each individual item.	Same as source file.
EXE (unsigned)	.NET assembly, Windows Forms application.	Satellite DLLs (.dll), one for each target language.
EXE (delay-signed)		Satellite DLLs (.dll), delay-signed.
EXE (strong name-signed)		Satellite DLLs (.dll), delay-signed*.
DLL (unsigned)	.NET assembly DLL, .NET Control library.	Satellite DLLs (.dll), one for each target language.
DLL (delay-signed)		Satellite DLLs (.dll), delay-signed.
DLL (strong name-signed)		Satellite DLLs (.dll), delay-signed*.
WPF BAML	Windows Presentation Foundation- text translation only (no GUI support).	Same as source file.

* A translated .NET satellite DLL for a **strong name-signed** .NET assembly (source) file is created by RC-WinTrans as a **delay-signed assembly**. Such a delay signed satellite DLL must ultimately be signed with the private key file (file extension: .snk) using the “SN.EXE” command line tool from Microsoft.

NOTE: RC-WinTrans also supports 64-bit (x64) application files.

3.1.3 Other File Types

FILE TYPE	DESCRIPTION	TEXT ENCODING (as of the March 2006 release)
PROPERTIES	Java properties file	ANSI/Escaped Unicode, MBCS, U1TF-8, UTF-16
INI	Windows INI file	ANSI/MBCS, UTF-8, UTF-16
SHL	InstallShield strings file	ANSI/MBCS, UTF-8, UTF-16
CSV	Tab-separated text file	ANSI/MBCS, UTF-8, UTF-16
PO	GNU PO file	ANSI/MBCS, UTF-8, UTF-16
XLIFF	XML (XLIFF 1.1) file	All encodings compatible with MSXML4 (ANSI, UTF-8, UTF-16 etc.)
<user-defined>	User-defined file formats. A user-provided, custom file converter can be seamlessly integrated into the RC-WinTrans localization environment as a VBA macro.	

3.1.4 XML Format Files

XML files come in a wide variety of types and structures. XML is no more than a set of basic rules for organizing almost any kind of data. Consequently the only way to properly support the localization of an XML file is to accommodate the file's specific internal structure and localizable contents.

RC-WinTrans 9 processes XML files using an XSL stylesheet to read and write a certain XML file format.

NOTE: RC-WinTrans does not have any commands or tools for defining the rules for processing an XML file. Only ready-made XML file parsers (XSL stylesheets) to enable the translation of various XML files are supported.

XML file converters can be easily added to the RC-WinTrans system where they are used whenever an XML file needs to be processed (read the source XML file and create the translated XML file).

How / Where Can XML Converters Be Obtained?

The XSL stylesheet must be created manually. This is a programming task that cannot be done with RC-WinTrans.

Some general knowledge about XML and XSL is necessary in order to realize the XSL stylesheet and also to build up the database file(s) that RC-WinTrans uses internally. Schaudin.com can make the required XML parser for your particular XML file(s) as an additional service. Please contact Schaudin.com for more information (see Chapter 7, "Contact and Support").

Advantages of Using XSL Stylesheets

1. Once you have a ready-made and adapted parser (XSL stylesheet) for your XML data, no other work is required to localize them with RC-WinTrans.
2. XSL stylesheets aren't as static as other file converters: it's always possible to modify the converter according to your needs. This way you have full control over how the XML files are processed.

Requirements for Translating XML Files

1. The XML files should contain elements with unique IDs to identify the elements. XML files that define the text content of a software GUI are ideal.
2. Plain XML text documents (without IDs) are **not** supported.

3.2 Translation Projects

Files that are to be translated are grouped into what is called a **translation project**. A translation project has a source language (the language of the project's source files) and one or more designated target languages.

The *Translation Project* symbol. Project file: "<projectname>**.rwtproject**."

3.2.1 What Is a Translation Project?

A translation project is an organizational unit that groups together **source files** needing to be translated, a **translation database file(s)** where the translation data is stored (single .fsmdb file or multiple .rwtxlf files, depending on the selected data storage type), and a **project organization file** (file extension: .rwtproject**)**.

A translation project stores all the localization information for the files contained in it: information about the files, the source and target languages, statistical information on the translation state of the project, etc. This information is physically stored in what is called a **project file**, described below.

Project Files

A project file (file extension: **.rwtproject**) contains the administration data of a project. RC-WinTrans creates and maintenances a single project file for each project. This file contains information such as the names of all the source files in the project, the target languages, and the name of the project. Project files are in the Windows INI file format.

Translation Database Files / Data Storage

A translation database file stores translation data in addition to supplemental data such as tasks, notes, and bookmarks. Translation data files can have one of two formats:

Database (file extension: .fsmdb): One database file containing the translation data of all the files in the project as well as all target language data- recommended.

XML (XLIFF) files (file extension: .rwtxlf): Translation data is stored in XML files, one for each source file and for each target language.

Database storage (.fsmdb) will be the preferred type for most users.

NOTE: The data storage type can be changed later on if for whatever reason this is desired. The command to convert a translation project is available in the *Tools* menu when all projects are closed.

Target Files
The target file is the ready-to-use, translated software file (resource or binary executable file) created with RC-WinTrans' *Write Target Files* command. A single source file can be represented by several different target files, with each one representing the source file content translated into a different target language.

3.2.2 Creating a Translation Project

In order to translate a file it must be part of an existing translation project. The New Translation Project Assistant guides you through the process of creating a new translation project and adding source files to it so that they may be translated with RC-WinTrans.

To create a new translation project, use the *New* | *Project* commands from the *File* menu (Figure 1).

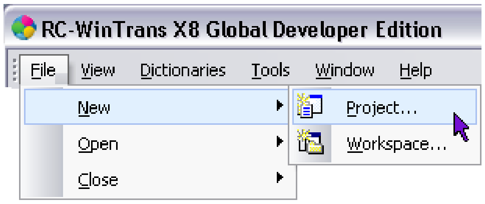


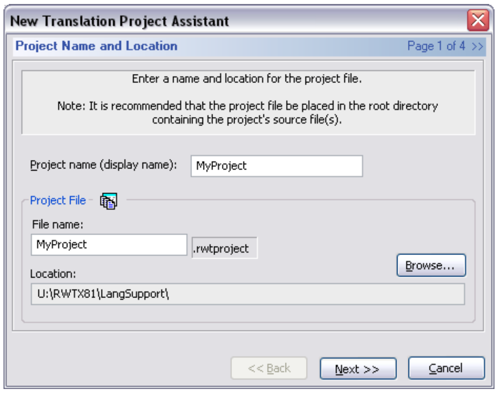
Figure 1: The menu commands used to create a new translation project with the New Translation Project Assistant.

New Translation Project Assistant
The New Translation Project Assistant will take you through four easy steps to create an RC-WinTrans translation project:

1. **Specify the name and location of the translation project.**
   The initial page of the New Translation Project Assistant (Figure 2) asks you to designate a name for the new project and select a location for it on your computer.

   Confirm your selection and continue on to the next step by pressing the *Next* button.

Figure 2: The *Project Name and Location* page (page 1 of 4) of the New Translation Project Assistant.



New Translation Project Assistant

Project Name and Location — Page 1 of 4 >>

Enter a name and location for the project file.

Note: It is recommended that the project file be placed in the root directory containing the project's source file(s).

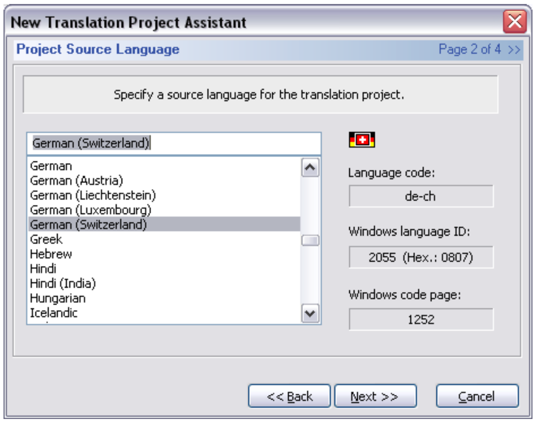
Project name (display name): MyProject

Project File

File name: MyProject .rwtproject

Location: U:\RWTX81\LangSupport\

2. **Specify the project's source language.**
   Select the project's source language from the list of languages on the *Project Source Language* page (Figure 3). This is the source language of the (source) file(s) that have been added to the project. After making your selection click the *Next* button to continue on to the next step.



New Translation Project Assistant

Project Source Language — Page 2 of 4 >>

Specify a source language for the translation project.

German (Switzerland)

- German
- German (Austria)
- German (Liechtenstein)
- German (Luxembourg)
- German (Switzerland)
- Greek
- Hebrew
- Hindi
- Hindi (India)
- Hungarian
- Icelandic

Language code: de-ch

Windows language ID: 2055 (Hex.: 0807)

Windows code page: 1252

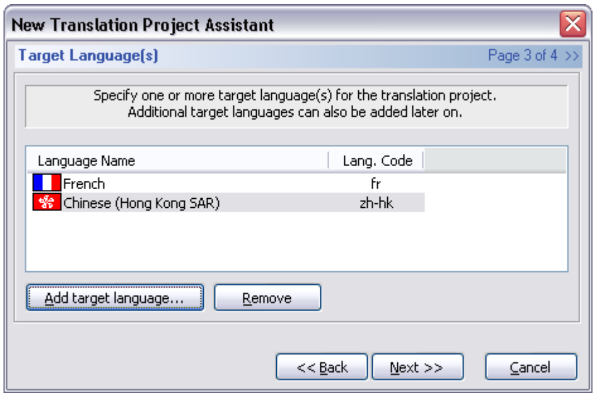
Figure 3: Specification of the new project's target language(s) on the *Project Source Language* page (page 2 of 4) of the New Translation Project Assistant.

3. **Specify the target language(s).**
The *Target Language(s)* page (Figure 4) asks you to select one or more target languages for the new project. Click on the *Add target language* button at the bottom of the page to call up the *New Target Language* dialog box to select the language that you want to designate as a target language from the list provided. Press the *OK* button to close the dialog box and add the language to the list of selected target languages.

A designated target language can be removed by selecting it in the list and clicking on the *Remove* button on the *Target Language(s)* page. Target languages can be added to/removed from an existing project at any time using the *Languages* page of the *Project Properties* dialog box. There are a few different ways to access this page- the simplest is via the *Languages* toolbar button at the top of the *Project* window.

The *Languages* toolbar button.

Confirm your choice(s) and continue on to the next step by pressing the *Next* button.



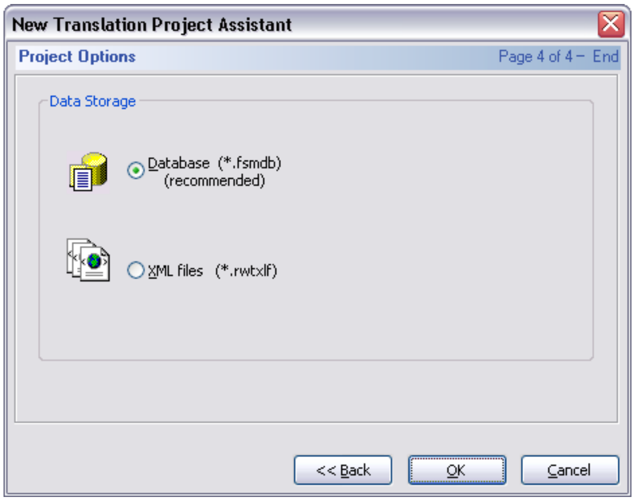
Specify one or more target language(s) for the translation project.
Additional target languages can also be added later on.

Language Name	Lang. Code
French	fr
Chinese (Hong Kong SAR)	zh-hk

Figure 4: The *Target Language(s)* page (page 3 of 4) of the New Translation Project Assistant.

4. **Select a data storage type for the new project.**
   The fourth and final step in creating a new translation project is to select the type of data storage to be used. Choose from a single database file (recommended) or multiple XML files.

   Press the *OK* button to close the Assistant and create the new project, which will be added to the *Projects Overview* tab view (*Workspace* bar) and automatically opened in a *Project* window (Section 3.3.1). The item in the tree view exists even when the project is closed and when RC-WinTrans is closed and then reopened. Double-click on the project item in the *Projects Overview* tab to quickly open the project the next time that you start RC-WinTrans.



New Translation Project Assistant

Project Options — Page 4 of 4 – End

Data Storage

● Database (*.fsmdb) (recommended)

○ XML files (*.rwtxlf)

Figure 5: The final page (*Project Options*) of the New Translation Project Assistant.

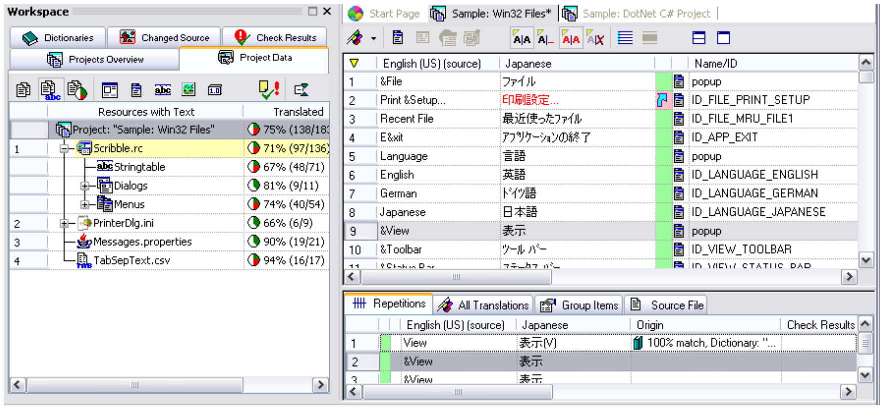
Adding Source Files to a Project

The newly-created translation project is initially empty- to add a (source) file(s), use the *Add File (to Project)* command (*File* menu or right mouse-click in the *Projects Overview* tab).

3.3 Translating a Project

3.3.1 The *Project* Window

The *Project* window is the main window in which a translation project is opened and its data is displayed. It is a MDI (multiple document interface) window, or document window. The *Project* window has a main view which is called the *Text Table* view (for its organization of data in table format) and four secondary (tab) views: the *Repetitions* view, the *All Translations* view, the *Group Items* view, and the *Source File* view.



	Resources with Text	Translated
	Project: "Sample: Win32 Files"	75% (138/18
1	Scribble.rc	71% (97/136)
	Stringtable	67% (48/71)
	Dialogs	81% (9/11)
	Menus	74% (40/54)
2	PrinterDlg.ini	66% (6/9)
3	Messages.properties	90% (19/21)
4	TabSepText.csv	94% (16/17)

	English (US) (source)	Japanese	Name/ID
1	&File	ファイル	popup
2	Print &Setup...	印刷設定...	ID_FILE_PRINT_SETUP
3	Recent File	最近使ったﾌｧｲﾙ	ID_FILE_MRU_FILE1
4	E&xit	ｱﾌﾟﾘｹｰｼｮﾝの終了	ID_APP_EXIT
5	Language	言語	popup
6	English	英語	ID_LANGUAGE_ENGLISH
7	German	ﾄﾞｲﾂ語	ID_LANGUAGE_GERMAN
8	Japanese	日本語	ID_LANGUAGE_JAPANESE
9	&View	表示	popup
10	&Toolbar	ﾂｰﾙ ﾊﾞｰ	ID_VIEW_TOOLBAR

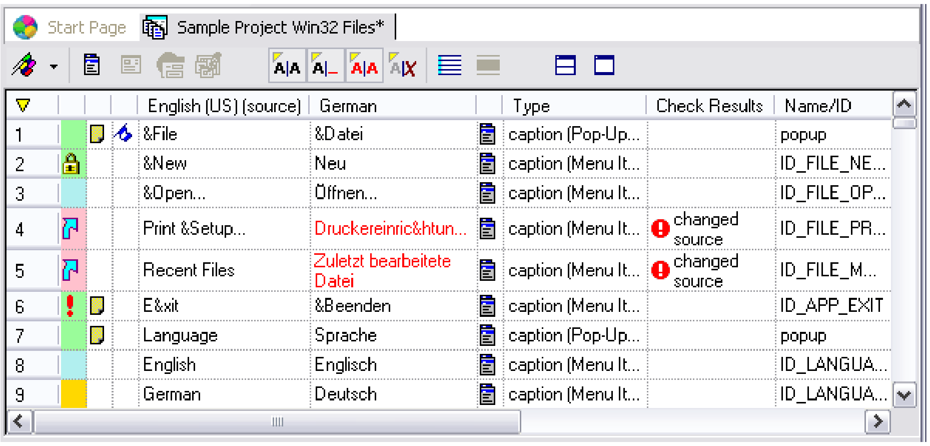
	English (US) (source)	Japanese	Origin	Check Results
1	View	表示(V)	100% match, Dictionary: "...	
2	&View	表示		
3	&View	表示		

Figure: A translation project opened in a *Project* window (right). The *Project Data* tab in the *Workspace* bar opened on the left shows the opened project (tree view item) and the source files contained therein. The data of the file selected there (“Scribble.rc”) is displayed in the *Text Table* view inside the *Project* window on the right. The *Repetitions* view is active below that in the *Project* window’s secondary view.

3.3.1.1 *Text Table* View (Primary View of a *Project* Window)

The *Text Table* view (Figure 1) lists (all or a selection of) the data contained in a translation project. By default only translation units that are text resource types (or string types) are listed, but translation units of all resource types, such as font, size and position data, and images can be shown in the *Text Table* view as well. All the most important information about a translation unit can be seen here at a glance, and the view’s interactive functions provide quick access to specific data and specialized context views. Most resources can be translated/edited directly in the *Text Table* view (inline editing).

Figure 1: RC-WinTrans' *Text Table* view. The *Project Window* toolbar is located directly above it.



▽	English (US) (source)	German	Type	Check Results	Name/ID
1	&File	&Datei	caption (Pop-Up...		popup
2	&New	Neu	caption (Menu It...		ID_FILE_NE...
3	&Open...	Öffnen...	caption (Menu It...		ID_FILE_OP...
4	Print &Setup...	Druckereinric&htun...	caption (Menu It...	❗ changed source	ID_FILE_PR...
5	Recent Files	Zuletzt bearbeitete Datei	caption (Menu It...	❗ changed source	ID_FILE_M...
6	E&xit	&Beenden	caption (Menu It...		ID_APP_EXIT
7	Language	Sprache	caption (Pop-Up...		popup
8	English	Englisch	caption (Menu It...		ID_LANGUA...
9	German	Deutsch	caption (Menu It...		ID_LANGUA...

Inline Editing

Translations can be entered directly in the *Text Table* view. Simply double-click into the appropriate cell of the *Target* column and type in the translation. Confirm a translation by pressing the "Return" key or by selecting another item.

Item Information

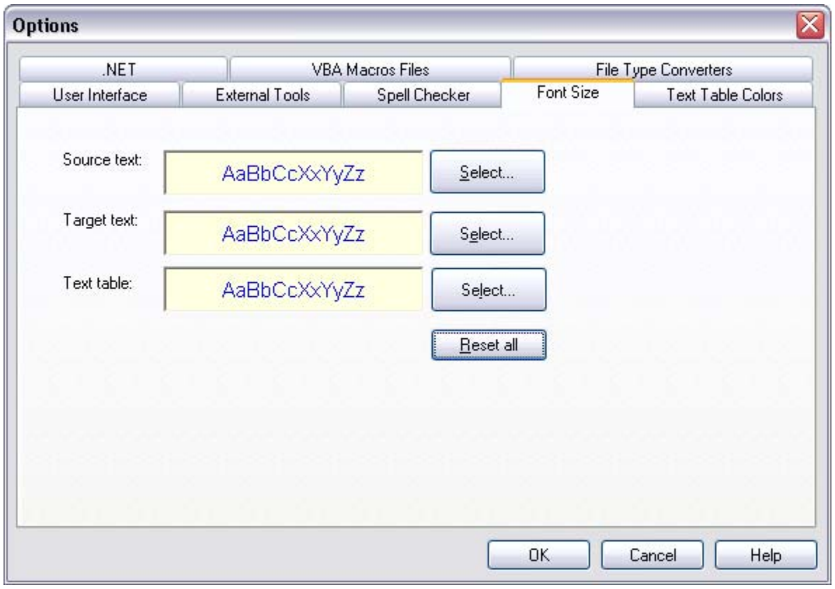
There can be various concurrent circumstances in a software translation project. For example, the project may need to be translated for the first time, or it may need to be updated because changes have been made to the project's source files. If changes have occurred, a previously-translated project may now contain new, non-translated items, or existing items may have been modified, possibly affecting their translations. There may be new files that have been added and that haven't been translated at all yet. Each individual item may also carry additional information, such as notes or markers. To handle all these variations and present them clearly to the user, RC-WinTrans uses a set of symbols in the *Text Table* view- these are described in the table that follows. An overview of all the symbols used in RC-WinTrans can be found in the RC-WinTrans Help under "Symbols."

SYMBOL	DESCRIPTION
Bookmark	Bookmarks are used to mark items so that they can easily be located amoung the contents of a translation project. See Section 3.5.5 of this guide for details.
! ! Re-check marker	The re-check marker is for items needing to be re-checked.
Task marker	Task markers appear next to items that have been assigned a task, or something that needs to be done to/for the item. Please refer to Section 3.5.5.
Note symbol	An item with the *Note* symbol has a note attached to it which can be viewed/edited in the *Note* field of the *Translation Edit* bar (Section 3.3.2). More information on notes can be found in Section 3.5.4 of this guide.
Unique Translation symbol	Designates a translation text as being “unique” (i.e., the text is translated differently than other instances of the same source text found elsewhere in the project). This keeps the item from being reported by the uniform translation check function.
🔒 *Locked* symbol ✗ *Excluded* symbol	Items bearing either of these symbols are to be skipped over in the translation process. A locked item cannot be edited and an item that has been excluded does not need to be translated.
❗ *Alert* symbol	This symbol marks items containing errors found by one of RC-WinTrans’ check functions (described in Section 3.5.1) or that have changed in the source file and need to be reviewed and/or modified.
✓ *Approved* symbol	Indicates that the state of the item has been set to “approved.”
Dialog Box symbol (open *Dialog Box* view) *Menu View* symbol (open *Menu* view)	The item can be viewed in its proper context/ WYSIWYG (**W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et) display is available for the item. Click on the respective symbol to open the *Dialog Box* or *Menu* view (described in Sections 3.3.4 and 3.3.6 of this guide).
Resource Type Editor symbol (open resource type editor (if available))	Other resource types may have a specialized editor available which can handle their modification (translation) and pass this information on to RC-WinTrans for inclusion in the target data.

Display Settings: Font Size and Style

The font size and style used for the display of texts in the *Text Table* view and the *Translation Edit bar* is set on the *Font Size* tab (Figure 2) located in the *Options* dialog box. Access this dialog by selecting the *Options* command from the *Tools* menu.

NOTE: A font (name, type) can be specified for each target language separately via a command button on the *Languages* tab in the *Project Properties* dialog box. The *Font Size* tab (Figure 2) is only used to specify the font size and the style (independent of a certain font type).



Source text: AaBbCcXxYyZz

Target text: AaBbCcXxYyZz

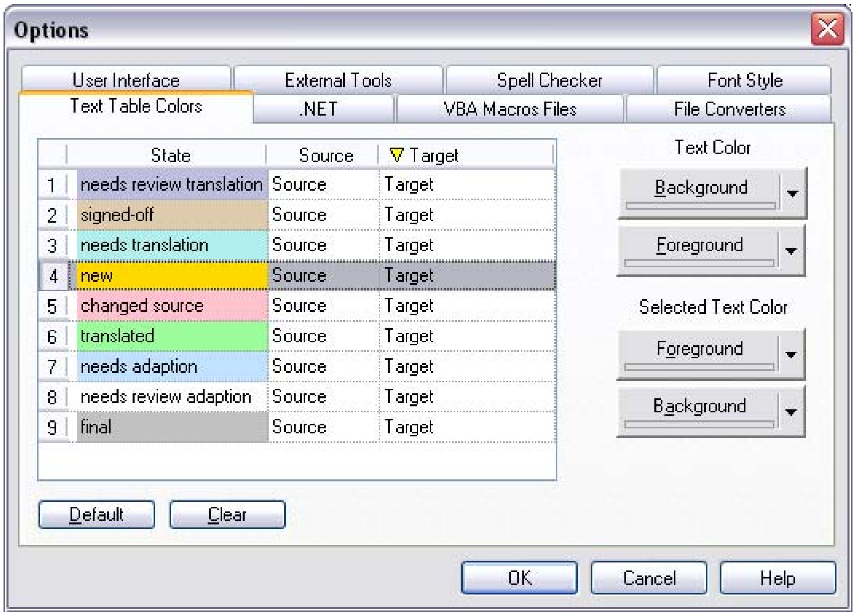
Text table: AaBbCcXxYyZz

Figure 2: The *Font Size* tab in the *Options* dialog box.

Display Settings: *Text Table* View Colors

The colors used for the font, cell backgrounds, and translation status indication in the *Text Table* view are also determined in the *Options* dialog box (*Text Table Colors* tab (Figure 3)). Click on the column for which you want to change the settings in this page’s interactive display, then select the appropriate button on the right to determine your color choices.

Figure 3: The *Text Table Colors* tab in the *Options* dialog box.



	State	Source	▽ Target
1	needs review translation	Source	Target
2	signed-off	Source	Target
3	needs translation	Source	Target
4	new	Source	Target
5	changed source	Source	Target
6	translated	Source	Target
7	needs adaption	Source	Target
8	needs review adaption	Source	Target
9	final	Source	Target

Display Settings: Display and Order of Columns in the *Text Table* View

The individual columns in the *Text Table* view can be arranged as desired by clicking on a column header with the (primary) mouse button and using the drag-and-drop mechanism (holding the button down while moving the mouse) to position it elsewhere.

Individual columns can be shown or hidden by selecting the *Text Table View* | *Show/Hide Columns* command from the *Window* menu to open the *Show/Hide Columns* dialog box. The same command is also available in a local menu called up by right-clicking in the *Text Table* view with the mouse. Check/Uncheck the boxes in the *Show/Hide Columns* dialog box to show/hide the corresponding column in the *Text Table* view.

The display of languages in the *Text Table* view is controlled elsewhere via the *Languages* toolbar button at the top of the *Project* window (Section 3.3.1.3). Clicking in a language's column makes that language the active target language (RC-WinTrans may take some time to switch over).

Data can be sorted alphabetically (source or target language) or according to name/ID, resource type, or translation status by clicking on the appropriate column header in the *Text Table* view.

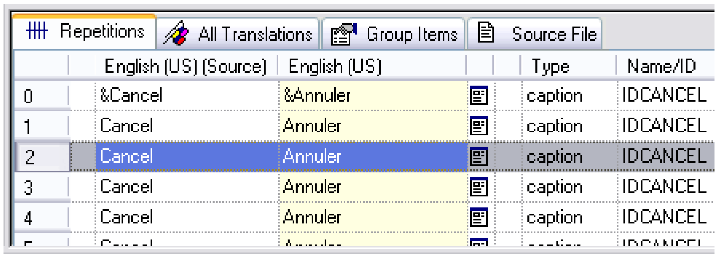
3.3.1.2 Secondary View of a *Project* Window (Tab Views)

The secondary view of the *Project* window is located below the primary (*Text Table*) view and consists of four different tabs, each one providing information on/assistance for a different aspect of the selected item and/or its localization. The secondary view can be shown or hidden by clicking on the *Tile Window* (shown left, top) or the *Maximize View* button (shown here, bottom) located in the *Project Window* toolbar (Section 3.3.1.3).

3.3.1.2.1 *Repetitions* Tab View

Translation units that have the exact same source text are considered repetitions. The *Repetitions* tab view lists all the repetitions for the item currently selected in the *Text Table* view. It has the same interactive capabilities as the *Text Table* view (e.g., inline editing, context view access).

NOTE: The *Repetitions* view lists all of the repetitions found in the translation project regardless of any filter which may currently be activated for the *Text Table* view. Repetitions are only shown for translation units that are string types. If a non-string type is currently selected in the *Text Table* view, the *Repetitions* view will be empty.



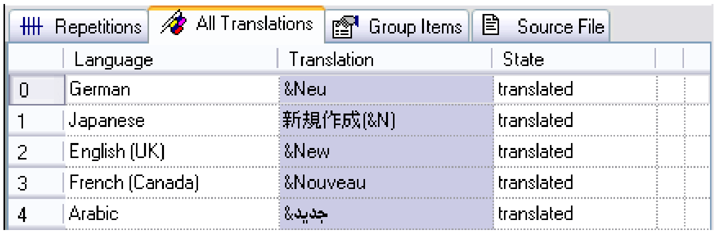
	English (US) (Source)	English (US)		Type	Name/ID
0	&Cancel	&Annuler	▤	caption	IDCANCEL
1	Cancel	Annuler	▤	caption	IDCANCEL
2	Cancel	Annuler	▤	caption	IDCANCEL
3	Cancel	Annuler	▤	caption	IDCANCEL
4	Cancel	Annuler	▤	caption	IDCANCEL
5	Cancel	Annuler	▤	caption	IDCANCEL

Figure: The *Repetitions* tab view active in the secondary view of a *Project* window.

3.3.1.2.2 *All Translations* View

The *All Translations* view shows the translation for the currently selected item in all the project's target languages. This view makes it possible to monitor and edit translations across different target languages without having to switch between (activate) different language pairs (source and target languages).

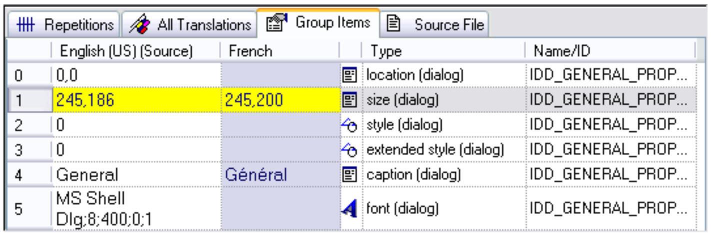
Figure: The *All Translations* tab view active in the secondary view of a *Project* window.



	Language	Translation	State
0	German	&Neu	translated
1	Japanese	新規作成(&N)	translated
2	English (UK)	&New	translated
3	French (Canada)	&Nouveau	translated
4	Arabic	&جديد	translated

3.3.1.2.3 *Group Items* View

The *Group Items* view lists all translation units belonging to the same group as the translation unit currently selected in the *Text Table* view. A “group” refers to a clustering of related data in the underlying database. For example, if the currently selected translation unit is a caption text of a dialog box item, the *Group Items* view will list all translation units that are in the same group, including translation units that are size, location, and style resource types.



	English (US) (Source)	French	Type	Name/ID
0	0,0		location (dialog)	IDD_GENERAL_PROP...
1	245,186	245,200	size (dialog)	IDD_GENERAL_PROP...
2	0		style (dialog)	IDD_GENERAL_PROP...
3	0		extended style (dialog)	IDD_GENERAL_PROP...
4	General	Général	caption (dialog)	IDD_GENERAL_PROP...
5	MS Shell Dlg;8;400;0;1		font (dialog)	IDD_GENERAL_PROP...

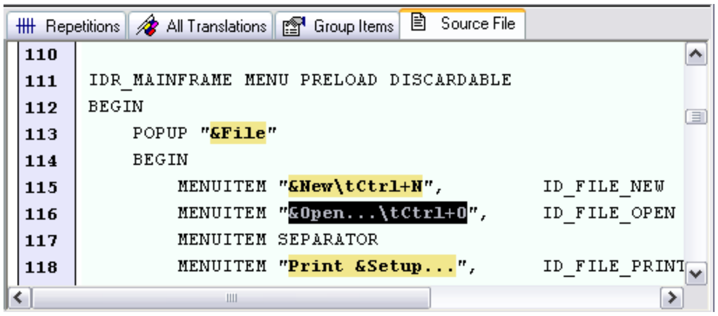
Figure: The *Group Items* tab view showing items from the same group (here: a dialog box) as the size data selected in the view.

3.3.1.2.4 *Source File* View

The *Source File* view (see figure) shows the formatting of the original source file (underlying data structure) with the currently selected text marked and other relevant items highlighted. Text items can be selected directly (using the mouse).

NOTE: This view can only be used for Win32 RC and Java properties files.

Figure: The *Source File* tab view showing the currently selected item (“&Open...\tCtrl+O”) of as it appears in the original file (here: a Win32 RC resources file).

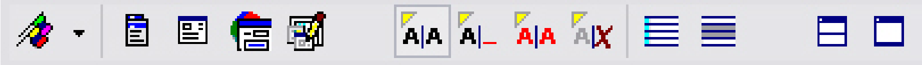


```

IDR_MAINFRAME MENU PRELOAD DISCARDABLE
BEGIN
    POPUP "&File"
    BEGIN
        MENUITEM "&New\tCtrl+N",        ID_FILE_NEW
        MENUITEM "&Open...\tCtrl+O",    ID_FILE_OPEN
        MENUITEM SEPARATOR
        MENUITEM "Print &Setup...",     ID_FILE_PRINT
```

3.3.1.3 *Project Window* Toolbar

The *Project* window has a toolbar at the top with selected commands for handling the project data. Brief descriptions of the toolbar buttons follow.



***Languages* Button**
Clicking on the arrow button on the right lets you change the active target language by selecting one from the list that appears. A click on the flags icon/left part of the button opens the *Languages* page in the *Project Properties* dialog box for more language settings including adding/removing target languages and controlling their visibility in the *Text Table* view.

***Open Menu View* Button**
Opens the *Menu* view (Section 3.3.6). Available only if the item currently selected in the *Text Table* view is a menu resource.

***Open Dialog Box View* Button**
Opens the *Dialog Box* view (Section 3.3.4). Available only if the item currently selected in the *Text Table* view is a dialog box resource.

***Open External .NET Forms Editor* Button**
Opens the External .NET Forms Editor (Section 3.3.5). Available only if the item currently selected in the *Text Table* view belongs to a .NET Form.

***Open Resource Type Editor* Button**
If available, displays a list of one or more resource type editors which can be used to modify the resource currently selected in the *Text Table* view.

***Show Translated Items* Button**
When activated, displays translated items in the *Text Table* view; hides them when not activated.

***Show Non-Translated Items* Button**
When activated, displays items having no translation text in the *Text Table* view; hides them when not activated.

***Show Items with Changed Source* Button**
When activated, displays items having the translation status "changed," (as a result of a database update to include changes made to the source file) in the *Text Table* view; hides them when not activated.

***Show Items Excluded from Translation* Button**
When activated, displays items that have been excluded from translation in the *Text Table* view; hides them when not activated.

***Show Marked Items Only* Button**
When activated, displays only items that are marked with a blue dot (or temporary) marker (e.g., from a find text operation) in the *Text Table* view; shows them listed among the other active items in the view when not selected.

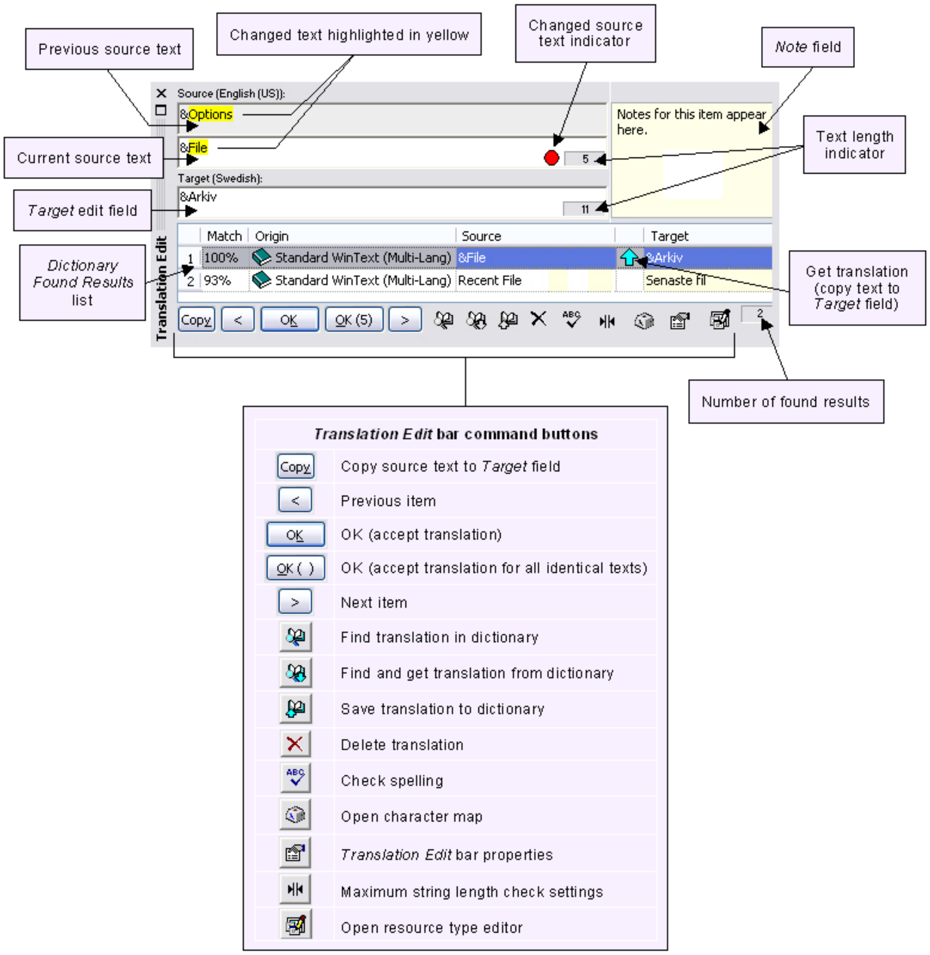
***Show Selected Items Only* Button**
When activated, displays only items that have been (multiple-) selected in the *Text Table* view; shows them listed among the other active items in the view when not selected.

***Tile Window* Button**
Splits the *Project* window into two views with the *Text Table* view on the top and the tab views situated below it.

***Maximize Text Table View* Button**
Fills the *Project* window with the primary (*Text Table*) view and hides the secondary view (tab views).

3.3.2 *Translation Edit* Bar

The *Translation Edit* bar is the primary place for translating and editing a selected translation unit. Several important commands and features are clustered in this area to make translation faster and easier. The following figure provides a graphical overview of the *Translation Edit* bar's features and functions for easy reference.



Source (English (US)):

&Options

&File

5

Notes for this item appear here.

Target (Swedish):

&Arkiv

11

	Match	Origin	Source	Target
1	100%	Standard WinText (Multi-Lang)	&File	&Arkiv
2	93%	Standard WinText (Multi-Lang)	Recent File	Senaste fil

Translation Edit bar command buttons	
Copy	Copy source text to *Target* field
<	Previous item
OK	OK (accept translation)
OK ( )	OK (accept translation for all identical texts)
>	Next item
	Find translation in dictionary
	Find and get translation from dictionary
	Save translation to dictionary
✕	Delete translation
✓	Check spelling
	Open character map
	Translation Edit bar properties
	Maximum string length check settings
	Open resource type editor

Figure: The *Translation Edit* bar.

Translation Edit Bar Commands and Features

- ***Changed Source Text Indication***
  Whenever an item is changed in the source file and the translation database is updated in RC-WinTrans, the previous source data is retained and displayed to the user in the *Translation Edit* bar. The previous source text is shown at the top of the *Translation Edit* bar, the current source text appears just below it. Changes are highlighted in yellow so that they can be quickly identified and used to determine the most appropriate translation.

- ***Spelling Error Indication***
  Words for which the spell checker has detected a potential spelling error appear in red font in the *Target* edit field. The automatic spell check must be enabled in order to activate this feature. See Section 3.5.1.1, "Spell Check," for more details.

- ***Leading/Trailing Spaces Indication***
  Leading and trailing spaces are indicated with a colored underline mark in the display area of the *Translation Edit* bar so that they can be seen and included in the translation text.

- ***Automatic Translation of Repetitions***
  When a source text is exactly the same as one or more other texts contained in the translation project, the number of repetitions is indicated in the parentheses of the *OK (__)* button. Pressing this button accepts the translation for the currently selected item and for all repetitions simultaneously.

- ***Finding Translations in Dictionaries***
  Search for translations in dictionaries that have been made available through the *Dictionaries* view (discussed in Sections 3.3.3.3 and 3.6.1 of this guide) by entering the source text in the *(Current) Source* field and pressing the Enter key. Select a portion of the text to search for just that part, or enter your own text to search for. Found translations are listed in the *Dictionary Found Results* list below the *Target* edit field.

- ***Dictionary Found Results List***
  This list displays the results of a *Find Translation (in Dictionaries)* operation. The text and its corresponding translation are shown along with the translation's match value (relative to the search string) and the dictionary in which the translation was found. To use a found translation, select it in the list and click on the blue arrow to insert the text into the *Target* field.

- ***Auto-Translate***
  Run the automatic translation function for the selected item using the command button located at the bottom of the *Translation Edit* bar. Available and selected resources will be searched to find the best matching translation for the source text. See Section 3.6.4.8 for more on this function.

- ***Character Map***
  Use the *Character Map* command button to select and insert a foreign character into the target text.

- ***Notes***
  A note can be attached to the current translation unit by entering text directly in the *Translation Edit* bar’s *Note* field. The *Note* field can be hidden and its size adjusted by clicking on the *Translation Edit Bar Properties* button.

- ***Image and Icon Translation***
  The *Translation Edit* bar shows image and icon resources and lets them be “translated” (edited) with an external graphics program. See Section 3.3.7, “Localizing Images” for further information.

- ***Resource Type Editors***
  RC-WinTrans lets you edit (“translate”) certain non-textual resource types in the most appropriate manner by allowing external programs to handle their modification. Once an external program, or resource type editor, has been assigned to handle a specific resource type, it can be accessed via the *Translation Edit* bar’s *Resource Type Editor* button. Multiple resource type editors can be made available for one and the same resource type where each may handle a different aspect of the translation unit.

- ***Quick Active Window Change***
  Use the Ctrl+I key combination to switch the input focus between the active *Project* window (the main *Text Table* view) and the *Translation Edit* bar’s *Target* edit field. When the focus is switched, this symbol will flash to ☞ indicate the active view/input window.

- ***Command Customization***
  You can determine whether pressing the Enter key will execute the *OK* or the *OK Next* command in the dialog box called up by clicking on the *Translation Edit Bar Properties* button. The *OK* command simply accepts the translation while the *OK Next* command accepts the translation and jumps to the next non-translated item.

3.3.3 The *Workspace* Bar

The RC-WinTrans *Workspace* bar features five detailed, interactive views to effectively manage translation data and ensure its accuracy with supportive tools and functions. Here you can do things like filter data according to resource type, view (source file) changes which may affect existing translation data, locate problems found by one or more of RC-WinTrans' check functions, and manage dictionaries/translation memories.

The *Workspace* bar's five tab views are:

1. The ***Projects Overview* tab**, providing an at-a-glance summary of the contents of translation projects and files.
2. The ***Project Data* tab**, showing detailed account of project contents with filtering options to isolate specific types of resources.
3. The ***Dictionaries* tab** with an overview of all available dictionaries (and/or Trados translation memories) and options for working with them.
4. The ***Changed Source* tab**, showing all data that has changed as a result of a database update operation.
5. The ***Check Results* tab**, listing the results of one or more check functions that have been executed.

3.3.3.1 *Projects Overview* Tab View

This tab organizes translation projects and the files they contain in a simple tree structure that gives users an at-a-glance summary of the contents of a selected workspace/project/file. The *Projects Overview* tab also provides access (toolbar/local menu) to certain project management commands such as adding/removing files and creating new translation projects.

When a translation project is opened, RC-WinTrans automatically detects whether changes have been made to any of the source files and indicates this to the user with a red dot on the project and the affected file(s). Please refer to Section 3.4 of this guide for more information on source file changes handling.

Bear in mind that this view is- as its name implies- for working with projects and the files that they contain very generally. For a more detailed display of this content plus advanced translation data filtering options, use the *Project Data* tab view (described in the next section).

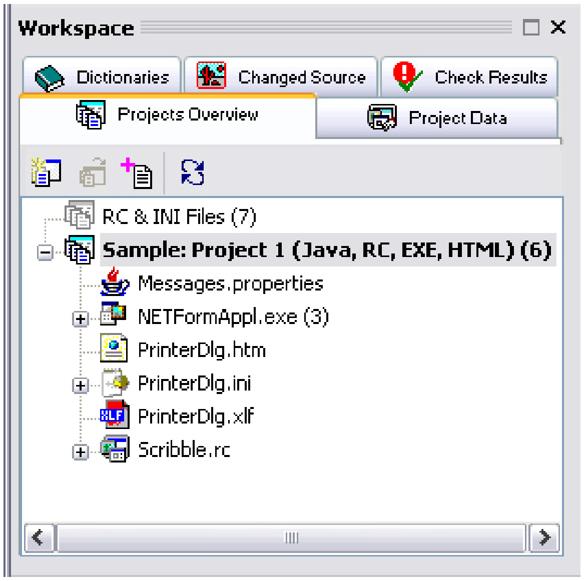


Figure: The *Workspace* bar's *Projects Overview* tab view outlines the contents of a translation project and provides access to selected, project management-type commands.

3.3.3.2 *Project Data* Tab View

The *Project Data* tab also shows the contents of a selected translation project, but the information shown here is much more detailed, and the commands it features are specifically designed to make working with/sorting through translation data faster and easier.

Data Filters

The tree view in the *Project Data* tab view lists a project's files and resources according to the active filter setting, determined by the filter toolbar button currently selected in the view. Filters are available to show only (i.e., isolate) specific resources types (e.g., dialogs, menus, or stringtable strings) or to show only resources that still have text to be translated, or to simply show the entire contents of the project.

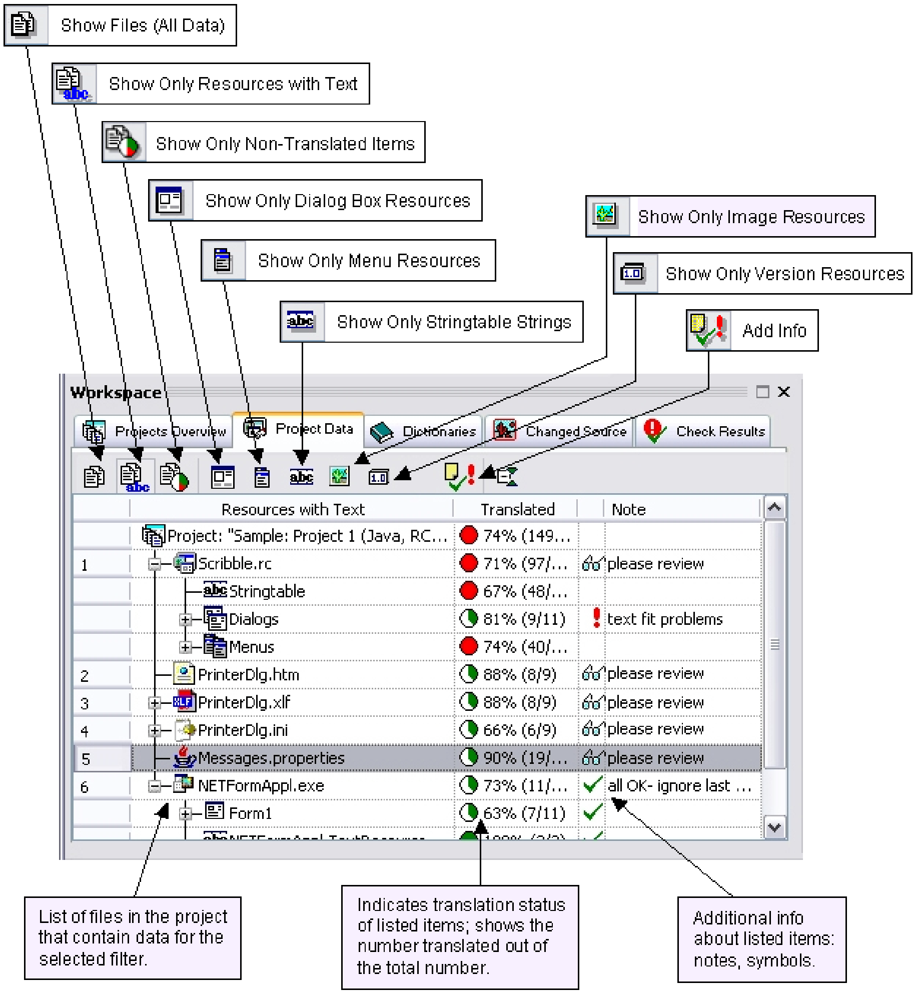
The main filter commands are available from the toolbar in the *Project Data* view. Additional filters are also available in a local menu called up with a right mouse-click in the view, and in the *Edit/View* menus and the *Project Window* toolbar.

Resource Type Filters

FILTER	DESCRIPTION
Files	Shows all files and all resources contained in the project.
Dialog box items	Shows all resources that are dialog box items.
Menu items	Shows all resources that are menu items.
Stringtable strings	Shows all resources that are stringtable strings.
Images	Shows all images in the project.
VersionInfo resources	(Win32) Shows all VersionInfo resources.
Accelerator tables	(Win32) Shows all accelerator tables.
File includes	(Win32) Shows all file includes.
Custom .NET types	.NET: Shows all non-standard (custom) resource types. (NOTE: Custom types are not initially known to (recognized by) RC-WinTrans as types to be translated. These types can be specified as text types (to be translated) or as “not localizable” in RC-WinTrans’ *Custom Resource Type* dialog box.)
Additional data	Additional data for dialog and menu resources can also be made visible if desired. This includes data such as the shortcut texts for menu items and the font, style, size, and position of dialog box items.
All resources with text	Shows only those resources that have text to translate.
All non-translated items	Shows only those resources that still need to be translated (or that have changed and need to have their translations confirmed).

The following figure shows a graphical overview of the *Project Data* view’s commands and information display for easy reference.

Figure: The *Project Data* tab view in the *Workspace* bar.


Show Files (All Data)
Show Only Resources with Text
Show Only Non-Translated Items
Show Only Dialog Box Resources
Show Only Menu Resources
Show Only Stringtable Strings
Show Only Image Resources
Show Only Version Resources
Add Info
Workspace
Projects Overview
Project Data
Dictionaries
Changed Source
Check Results

	Resources with Text	Translated	Note
	Project: "Sample: Project 1 (Java, RC…	● 74% (149…	
1	Scribble.rc	● 71% (97/…	please review
	Stringtable	● 67% (48/…	
	Dialogs	◑ 81% (9/11)	! text fit problems
	Menus	● 74% (40/…	
2	PrinterDlg.htm	◑ 88% (8/9)	please review
3	PrinterDlg.xlf	◑ 88% (8/9)	please review
4	PrinterDlg.ini	◑ 66% (6/9)	please review
5	Messages.properties	◑ 90% (19/…	please review
6	NETFormAppl.exe	◑ 73% (11/…	✓ all OK- ignore last …
	Form1	◑ 63% (7/11)	✓

List of files in the project that contain data for the selected filter.

Indicates translation status of listed items; shows the number translated out of the total number.

Additional info about listed items: notes, symbols.

Translation Status Indication

Each item listed in the *Project Data* tab has a translation status indicator which tells the user how much of a given resource/resource type/file/project is translated. This is expressed in a percentage next to a pair of numbers which is the number of translated items followed by the total number of items to be translated. The colored circles signify translation (green), incomplete transla-tion (white), and changed (new/modified/deleted resources) source data (red), and may be filled in with one or a combination of these colors.

NOTE: RC-WinTrans considers an item translated when its state is set to either “translated” or “final.”

State Symbols, Re-Check Markers and Notes

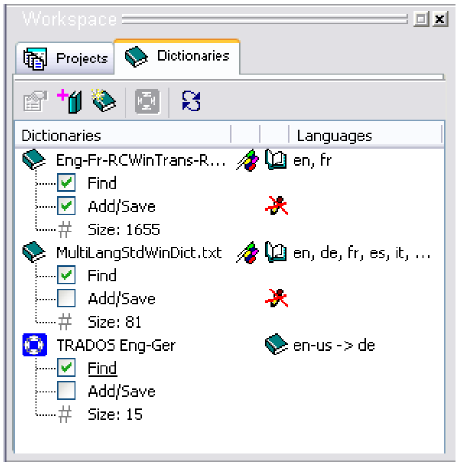
Supply additional information about items listed in the *Project Data* view with one of two “state” symbols, or statuses (“Approved” and “For Review”), a re-check marker and/or a note. These are displayed next to their respective items in the view. Assign a note, marker, or flag using the *Add Info* toolbar button, or by right mouse-clicking on the tree item to open a local menu with the same commands.

Interactive Display

The *Project Data* view lists the entire contents of a translation project in an interactive display (tree view) that lets you jump to (select) specific data in RC-WinTrans’ *Text Table* view (described in Section 3.3.1.1 of this guide) simply by clicking on the corresponding item listed in the view. You will notice that the item currently selected in the *Text Table* view always corresponds to the one selected in the *Project Data* tab. For example, when a project is opened the “Project” item at the top of the tree view (the root item) is selected and the *Text Table* view shows all the data contained in the project. If a menu or a dialog box is selected in the view, only the data for that selected item will be shown in the *Text Table* view. Click on the “Project” item and all the project data will be visible again.

3.3.3.3 *Dictionaries* Tab View

The *Dictionaries* tab view displays a list of dictionaries and/or Trados™ translation memories (TMs) that have been made available to RC-WinTrans to support the translation process. Manage dictionaries/TMs with commands to create new, add existing, and remove registered dictionaries.



Dictionaries	Languages
Eng-Fr-RCWinTrans-R...	en, fr
☑ Find	
☑ Add/Save	
# Size: 1655	
MultiLangStdWinDict.txt	en, de, fr, es, it, ...
☑ Find	
□ Add/Save	
# Size: 81	
TRADOS Eng-Ger	en-us -> de
☑ Find	
□ Add/Save	
# Size: 15	

Figure: The *Dictionaries* tab view showing two (text file) dictionaries and a Trados translation memory registered to support the translation of text resources.

Information shown in the Dictionaries view next to the dictionary/TM items listed there indicates:

- The languages that the dictionary/TM supports.
- Whether the dictionary/TM supports the same (source and target) languages active in the translation project (*Language Support* symbol:). A dictionary with language support is considered "active," meaning that it can be used to find and get translations.
- Whether the dictionary/TM is currently in use (*Dictionary in Use* symbol:) or not in use (*Dictionary* symbol:).
- Whether the dictionary/TM is write-protected ("read-only"-).
- Whether the dictionary/TM is enabled as a "find reference" or a "save reference" (used to search for/save translations, respectively).
- The size of the dictionary/TM (number of entries).
- Whether the Trados Translator's Workbench is currently running.

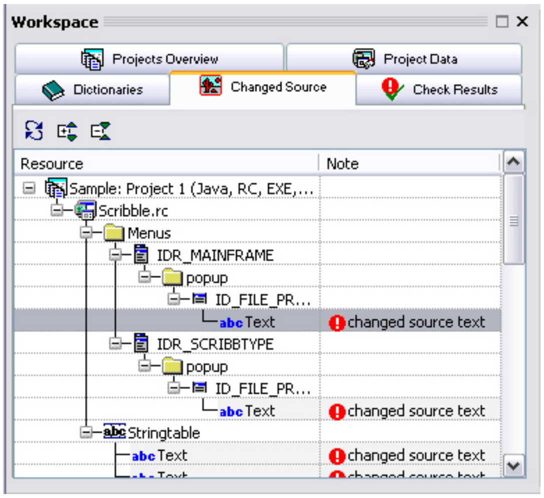
More information on the *Dictionaries* tab view can be found in Section 3.6.1 of this guide.

3.3.3.4 *Changed Source* Tab View

The *Changed Source* tab view lists all resources affected by changes made to the source file. After running the *Update Database* command, all affected resources will be listed in the view. Clicking on an item in the list will select it in both the *Text Table* view and the *Translation Edit* bar where it can be translated, reviewed and/or modified. Once the translation text has been confirmed, the state of the item changes to "translated" and all markers that distinguished it as a changed resource are removed. Confirming the translation will also remove the item from the *Changed Source* view.

NOTE: The changes that are detected by RC-WinTrans are not limited to text items. All localizable data is considered and changes thereto indicated in this view. Very often you will find that the changes include modifications made to the source size and/or position data of a dialog box and/or dialog box controls.

Selecting an item in the *Changed Source* view will automatically direct you to (open and select the item in) the most appropriate view for handling that particluar type of resource. This can be either the *Text Table* view, the *Dialog Box* view, or the *Menu* view. Whenever an item is selected in the *Text Table* view (as a result of being selected in the *Changed Source* tab) it will be shown together with the other changed items, replacing the data that was listed there previously.



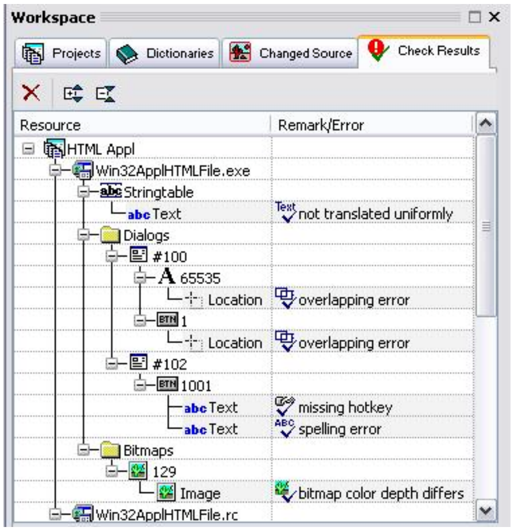
Resource	Note
Sample: Project 1 (Java, RC, EXE,...	
Scribble.rc	
Menus	
IDR_MAINFRAME	
popup	
ID_FILE_PR...	
abc Text	changed source text
IDR_SCRIBBTYPE	
popup	
ID_FILE_PR...	
abc Text	changed source text
abc Stringtable	
abc Text	changed source text
abc Text	changed source text

Figure: The *Workspace* bar's *Changed Source* tab view with a list of the changed items found during the update process.

Section 3.4 of this guide ("Source File Changes Handling") should be consulted for more information on source file changes, updating the translation database, and changes indication.

3.3.3.5 *Check Results* Tab View

The *Check Results* tab view is where errors (or potential errors) are listed after one or more of RC-WinTrans' check functions has been performed. Clicking on an entry in the list instantly selects that item in the most appropriate view (*Dialog Box*/*Menu*/*Text Table* view/*Translation Edit* bar) for evaluating/
correcting the found problem.



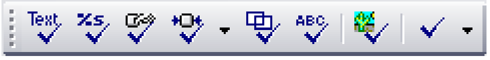
Resource	Remark/Error
HTML Appl	
Win32ApplHTMLFile.exe	
Stringtable	
abc Text	not translated uniformly
Dialogs	
#100	
A 65535	
Location	overlapping error
BTN 1	
Location	overlapping error
#102	
BTN 1001	
abc Text	missing hotkey
abc Text	spelling error
Bitmaps	
129	
Image	bitmap color depth differs
Win32ApplHTMLFile.rc	

Figure: The *Workspace* bar's *Check Results* tab view displaying a list of (possible) errors found by various check functions (uniform translation, overlapping items, missing/doubled hotkeys, etc.).

Running Checks

There are three ways to access (run) a check function:

1. The *Checks* toolbar.



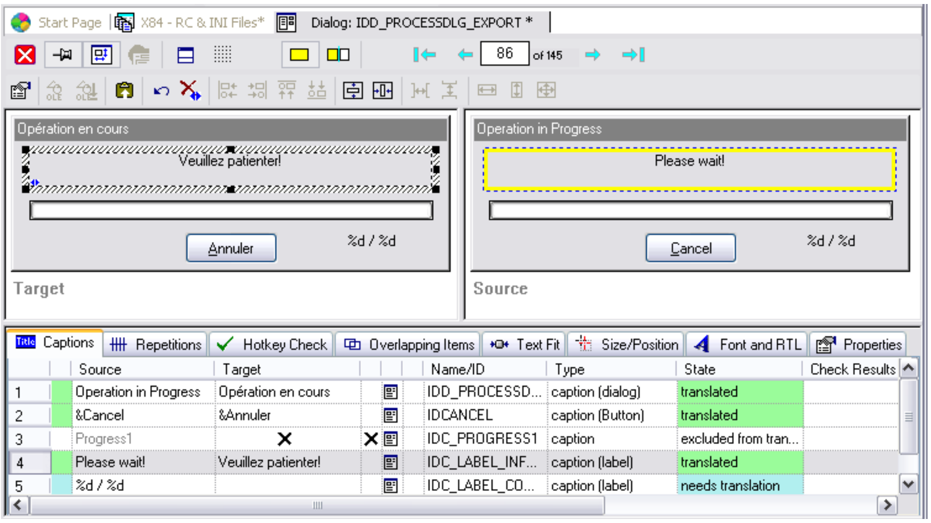
2. The *Checks* menu in the main menu bar.
3. Local menu opened with a right mouse-click in the view.

Section 3.5.1 of this guide deals with check functions and the sections on the *Dialog Box* (3.3.4) and *Menu* (3.3.6) views can assist with correcting problems found for those resource types.

3.3.4 *Dialog Box* View and Dialog Box Editor

The *Dialog Box* view shows a single dialog box resource as it appears in the source and target files. The text of dialog box captions and control items can be selected for translation/editing in this view and the Dialog Box Editor (accessed via the *Layout Editing* toolbar button) lets you change the size of a dialog box as well as the size and position of control items.

Figure: The *Dialog Box* view with two toolbars located above the display for source (right) and target (left) dialog boxes. The lower half of the window has a number of tab views displaying various information about the selected dialog box. Note that the *Layout Editing* toolbar button has been pressed to enable the Dialog Box Editor.



		Source	Target			Name/ID	Type	State	Check Results
1	🟩	Operation in Progress	Opération en cours			IDD_PROCESSD...	caption (dialog)	translated	
2	🟩	&Cancel	&Annuler			IDCANCEL	caption (Button)	translated	
3		Progress1	✕	✕		IDC_PROGRESS1	caption	excluded from tran...	
4	🟩	Please wait!	Veuillez patienter!			IDC_LABEL_INF...	caption (label)	translated	
5		%d / %d				IDC_LABEL_CO...	caption (label)	needs translation	

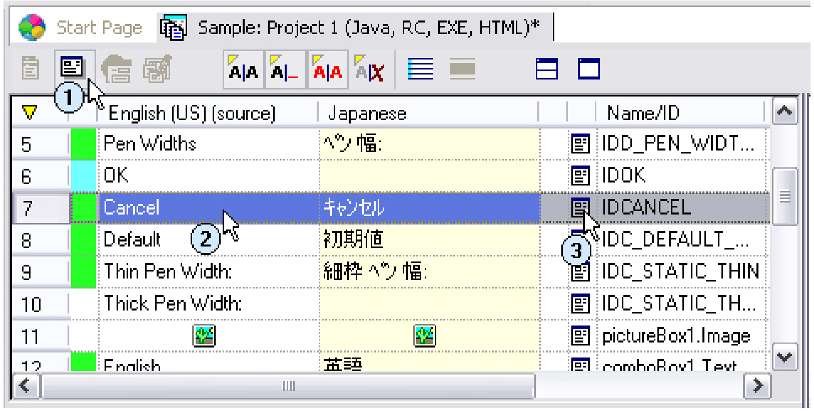
The *Dialog Box* view is used to display and edit **Windows Win32 dialog boxes** and **.NET Windows Forms**. RC-WinTrans also has a special (external) editor for .NET Windows Forms which can be opened by pressing the *External Editor* toolbar button (shown below).

The *Open External .NET Forms Editor* toolbar button.

3.3.4.1 Opening the *Dialog Box* View

The *Dialog Box* view can be opened from the main window of a translation project (*Project* window) in one of three ways:

1. Click on the *Open Dialog Box View* toolbar button.
2. Double-click on the entry in the *Text Table* view.
3. Click on the *Dialog Box* symbol in the *Text Table* view.



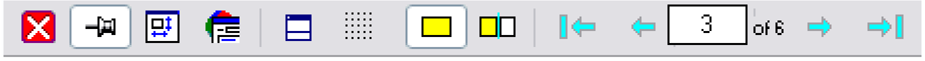
	English (US) (source)	Japanese	Name/ID
5	Pen Widths	ペン幅:	IDD_PEN_WIDT...
6	OK		IDOK
7	Cancel	キャンセル	IDCANCEL
8	Default	初期値	IDC_DEFAULT_...
9	Thin Pen Width:	細枠 ペン幅:	IDC_STATIC_THIN
10	Thick Pen Width:		IDC_STATIC_TH...
11			pictureBox1.Image
12	English	英語	comboBox1.Text

Figure: The three ways of opening the *Dialog Box* view described above.

3.3.4.2 *Dialog Box* View Toolbars

The *Dialog Box* view has two toolbars with shortcuts to various display and navigation commands as well as access to the layout functions for the Dialog Box Editor. Brief descriptions of the toolbar buttons follow.

***Navigation & Display* Toolbar**



***Close Window* Button**
Closes the *Dialog Box* view.

***Fixed Menu/Dialog Box View* Button**
Fixes, or designates, one view for displaying menu resources and another view for displaying dialog box resources. If this button is **not** activated, a new, separate view will be opened for each menu or dialog box that you want to display.

***Layout Editing* Button**
Activates the Dialog Box Editor to resize/reposition dialog boxes or control items.

***Open External .NET Forms Editor* Button**
Opens the External .NET Forms Editor (Section 3.3.5).

***Tile Window* Button**
Tiles the window vertically with the WYSIWYG (**W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et) display on top and the tab views at the bottom.

***Toggle Grid* Button**
Toggles the layout grid on and off.

***Source and Target* Button**
Displays the source and target dialog boxes side by side.

***Target View* Button**
Shows the currently selected dialog box as it will appear in the target file. The target dialog box display will be hidden.

***First Dialog* Button**
Selects and displays the first dialog box contained in the project.

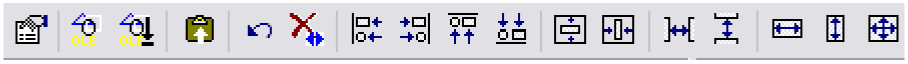
***Previous Dialog Box* Button**
Selects and displays the previous dialog box in the project.

Counter
Shows which dialog box is currently selected and the total number of dialog boxes contained in the project. Type a number in the field and press the Return/Enter key to select and display that number dialog box.

***Next Dialog* Button**
Selects and displays the next dialog box in the project.

***Last Dialog* Button**
Selects and displays the last dialog box contained in the project.

Layout Toolbar



Properties **Button**
Opens the *Properties* dialog box (for the dialog box or a selected control item).

Edit ActiveX Control Properties **Button**
Opens the property pages of a selected ActiveX control (Win32 software only). For a more in-depth description of ActiveX control handling see the RC-WinTrans Help topic “Localizing ActiveX Control Data.”

Reset ActiveX Control Properties **Button**
Resets the property data of a selected ActiveX control to the original source property data (Win32 software only).

Copy to Clipboard **Button**
Shows the current source and target dialog boxes side by side.

Undo Size/Position Changes **Button**
Undoes the last size and/or position changes made to the dialog box or dialog box item.

Set Original Size and Position **Button**
Resets the size and position of the selected dialog box and its control(s) to the original (source) settings.

Align Left **Button**
Aligns the left edge of the selected control with the dominant control.

Align Right **Button**
Aligns the right edge of the selected control with the dominant control.

Align Top **Button**
Aligns the bottom edge of the selected control with the dominant control.

Align Bottom **Button**
Aligns the bottom edge of the selected control with the dominant control.

***Center Horizontally* Button**
Centers the control horizontally within the dialog box.

***Center Vertically* Button**
Centers the control vertically within the dialog box.

***Space Horizontally* Button**
Evenly spaces the selected controls horizontally.

***Space Vertically* Button**
Evenly spaces the selected controls vertically.

***Same Width* Button**
Resizes the selected control to the same width as the dominant control.

***Same Height* Button**
Resizes the selected control to the same height as the dominant control.

***Same Size* Button**
Resizes the selected control to the same size as the dominant control.

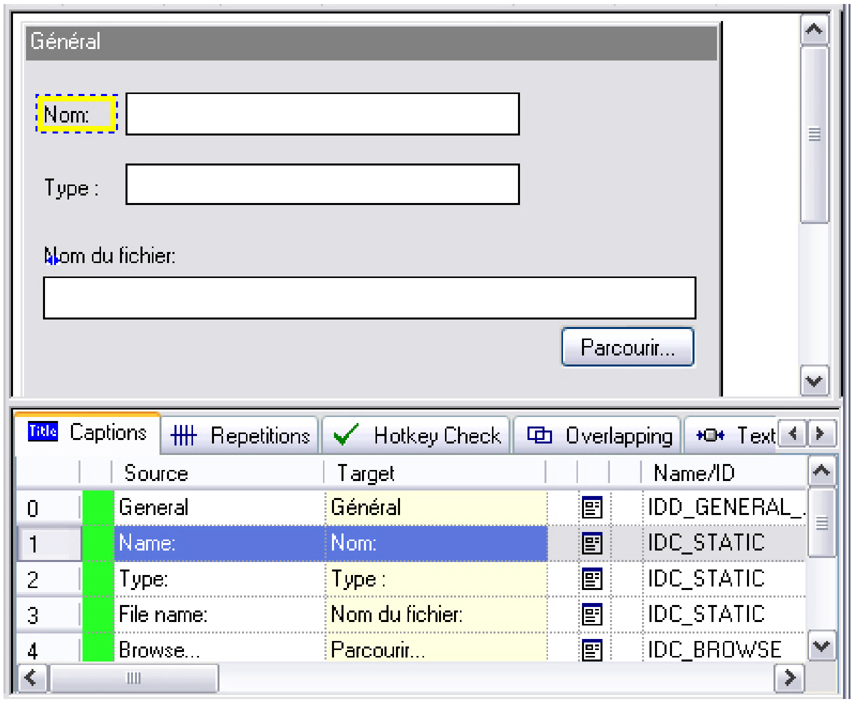
NOTE: Multiple items must be selected in order to activate the layout toolbar buttons (*Align Left* through *Same Size* buttons). Select multiple items by holding the “Shift” key down while clicking on items with the mouse.

3.3.4.3 Tab Views

The *Dialog Box* view has a secondary window with several different tab views for working with dialog boxes and dialog box elements. Each one handles a different aspect to ensure that these resource types are localized correctly.

3.3.4.3.1 *Captions* Tab View

The *Captions* tab lists the caption texts of all the controls in the currently selected dialog box. As with the *Text Table* view, an item listed in the *Captions* tab view can be selected and translated in the *Translation Edit* bar, or it can be edited directly in the table (inline editing).

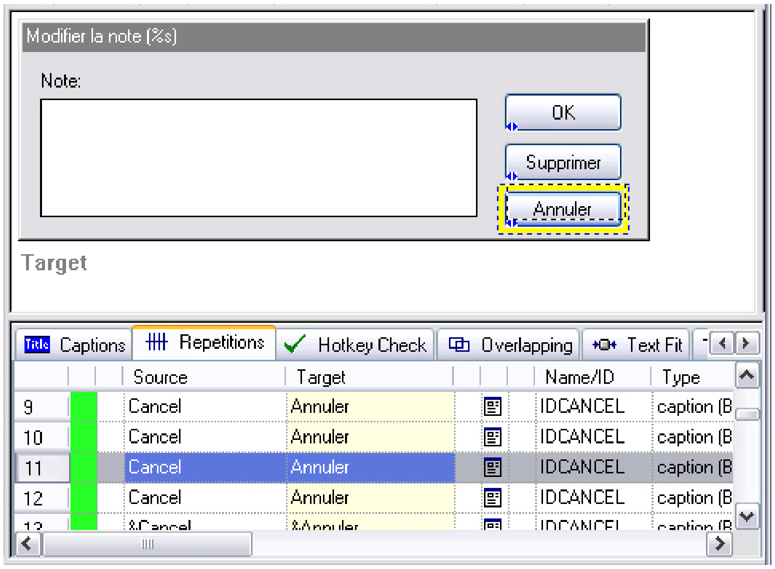


		Source	Target		Name/ID
0	🟩	General	Général		IDD_GENERAL_
1	🟩	Name:	Nom:		IDC_STATIC
2	🟩	Type:	Type :		IDC_STATIC
3	🟩	File name:	Nom du fichier:		IDC_STATIC
4	🟩	Browse...	Parcourir...		IDC_BROWSE

Figure: The *Dialog Box* view with the WYSIWYG display on the top and the *Captions* tab view active below it. The *Captions* tab lists all the caption texts from the currently selected dialog box.

3.3.4.3.2 *Repetitions* Tab View

The *Repetitions* tab view lists all occurrences of texts which are identical to the text of the item currently selected in the *Dialog Box* view. All translation units that have the same source text are considered repetitions. The view supports inline editing just like the *Text Table* view does.



		Source	Target		Name/ID	Type
9	🟩	Cancel	Annuler		IDCANCEL	caption (B
10	🟩	Cancel	Annuler		IDCANCEL	caption (B
11	🟩	Cancel	Annuler		IDCANCEL	caption (B
12	🟩	Cancel	Annuler		IDCANCEL	caption (B
13	🟩	&Cancel	&Annuler		IDCANCEL	caption (B

Figure: The *Dialog Box* view with the *Repetitions* tab showing multiple repetitions for the selected text “Cancel.”

3.3.4.3.3 *Hotkey Check* Tab View

The *Hotkey Check* tab lists all caption texts with missing or doubled hotkeys. The hotkey check and listing of hotkey errors are performed automatically- there is no need to run a check manually. The absence/presence of hotkey errors is indicated by one of two different symbols next to the tab’s title. symbols next to the tab title.

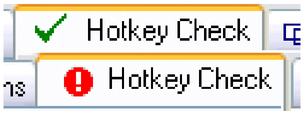
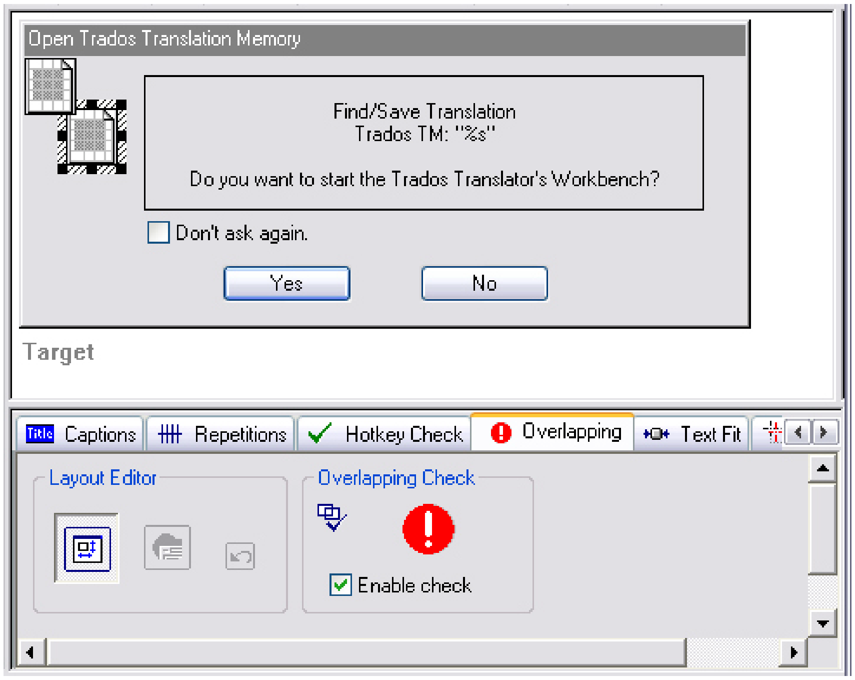


Figure: The tab of the *Hotkey Validation* tab view. A green check mark indicates that the dialog box contains no hotkey conflicts (no missing/doubled hotkeys). A red circle with exclamation mark (*Alert* symbol) indicates that the dialog box contains hotkey conflicts needing to be resolved. The affected control items are listed in the view.

The hotkey check and hotkey errors are described along with the other RC-WinTrans check functions in Section 3.5.1 of this guide.

3.3.4.3.4 *Overlapping* Tab View

The *Overlapping* tab alerts the user to any problems found by the overlapping items check and provides access to the appropriate layout editor (Dialog Box or .NET Form) to resize affected items. There are also options for undoing size/position changes (made since the layout was last saved).



Open Trados Translation Memory

Find/Save Translation
Trados TM: "%s"

Do you want to start the Trados Translator's Workbench?

□ Don't ask again.

☑ Enable check

Figure: The *Dialog Box* view showing the WYSIWYG display of a target dialog box at the top and the *Overlapping* tab active below it. The tab alerts the user to overlapping items in the dialog.

3.3.4.3.5 *Text Fit* Tab View

The *Text Fit* tab indicates whether the text of the currently selected dialog box item fits inside the borders of its control. The tab also has commands for resolving the text fit problem and an option for disabling the check for the selected control- please see the RC-WinTrans Help for more information on these.

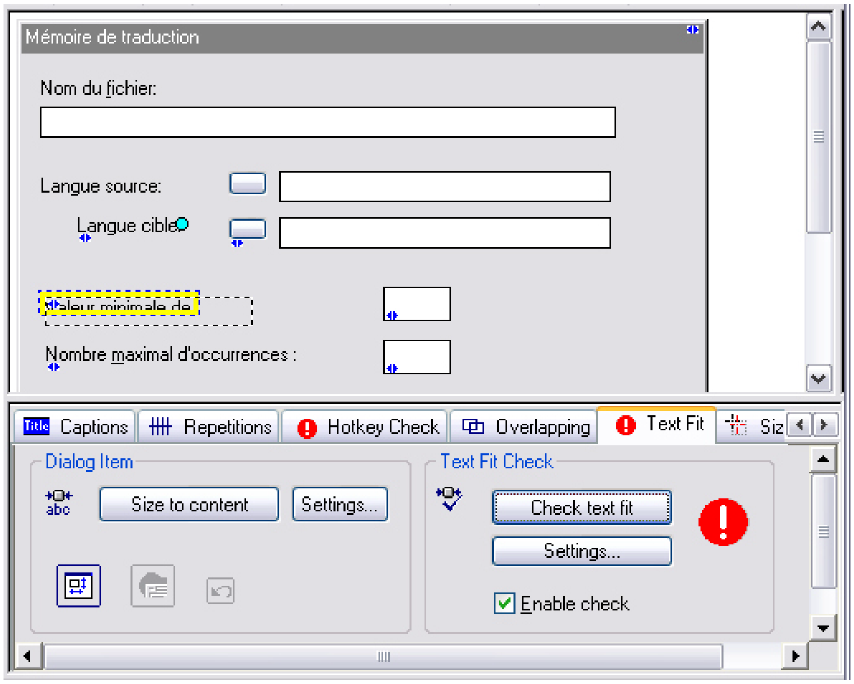
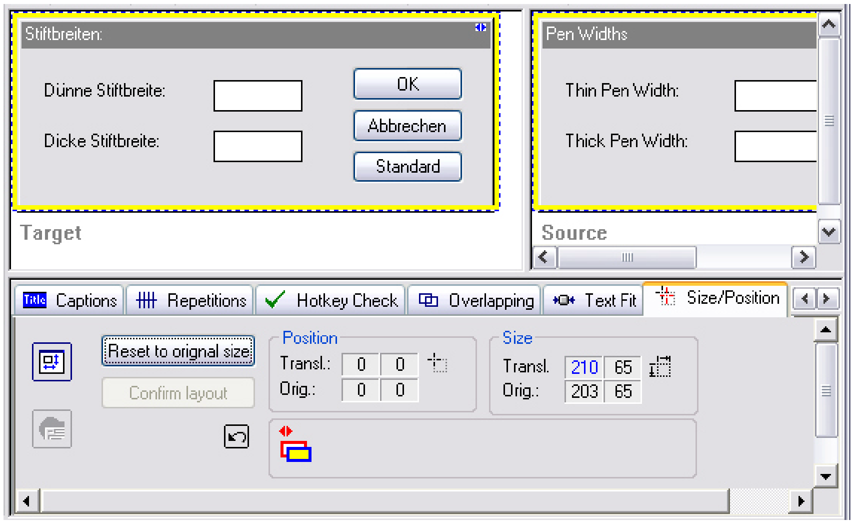


Figure: The *Dialog* Box view with the *Text Fit* tab showing a text fit problem for the currently selected item.

3.3.4.3.6 *Size/Position* Tab View

The *Size/Position* tab displays information about layout changes that have been made to a dialog box or dialog box item and provides options for modifying/resetting the layout. This is useful if layout changes have been made in the source file, making it necessary to check the size and position of the target dialog box control to see if it is still valid or if it needs to be updated.

Additional details can be found in the RC-WinTrans Help ("*Size/Positon* Tab View").



Position
Transl.: 0 0
Orig.: 0 0

Size
Transl. 210 65
Orig.: 203 65

Figure: The *Size/Position* tab active at the bottom of the *Dialog Box* view where the size of the selected dialog box has been modified ("Size" values).

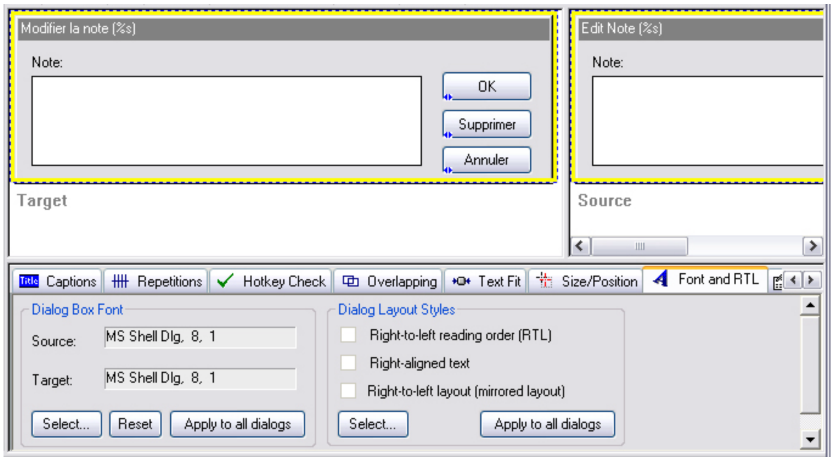
3.3.4.3.7 *Dialog Font and RTL* Tab View

The *Dialog Font and RTL* tab view is used to set the font of the current target dialog box and/or to set the font for all dialog boxes within the current translation project (*Apply to all dialogs* command button).

This view is also used to set the RTL (right-to-left) reading order for the current dialog box. As with the font, RTL flags can be set for the current dialog box alone or for all dialog boxes at once using the *Apply to all dialogs* button.

NOTE – WIN32 DIALOG BOXES VS. .NET FORMS: Dialog layout styles differ depending on whether the selection is a Win32 dialog box or a .NET Windows Form. The *Apply to all dialogs* commands will only work on dialog boxes of the same type (all other Win32 dialog boxes or all other .NET Windows Forms).

Figure 1: The *Dialog Box* view with the WYSIWYG display on top and the *Font and RTL* tab view active at the bottom. A **Win32 dialog box** is shown here.



Dialog Box Font

Source: MS Shell Dlg, 8, 1

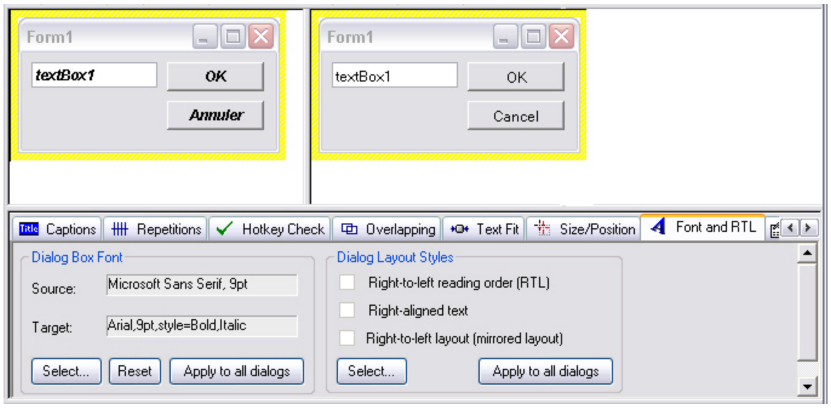
Target: MS Shell Dlg, 8, 1

Dialog Layout Styles

☐ Right-to-left reading order (RTL)

☐ Right-aligned text

☐ Right-to-left layout (mirrored layout)



Dialog Box Font

Source: Microsoft Sans Serif, 9pt

Target: Arial,9pt,style=Bold,Italic

Dialog Layout Styles

☐ Right-to-left reading order (RTL)

☐ Right-aligned text

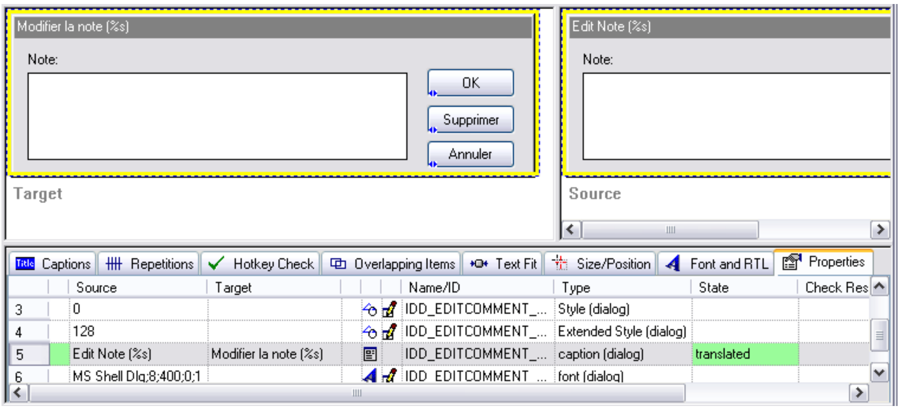
☐ Right-to-left layout (mirrored layout)

Figure 2: The *Dialog Box* view with the WYSIWYG display on top and the *Font and RTL* tab view active at the bottom. A **.NET Windows Form** is shown here.

3.3.4.3.8 *Properties* Tab View

The *Properties* tab view shows all the modifiable (translatable) properties of the currently selected dialog box control or of the dialog box itself.

Figure: The *Dialog* Box view with the *Text Fit* tab showing a text fit problem for the currently selected item.



		Source	Target			Name/ID	Type	State	Check Res
3		0				IDD_EDITCOMMENT_...	Style (dialog)		
4		128				IDD_EDITCOMMENT_...	Extended Style (dialog)		
5		Edit Note (%s)	Modifier la note (%s)			IDD_EDITCOMMENT_...	caption (dialog)	translated	
6		MS Shell Dlg;8;400;0;1				IDD_EDITCOMMENT_...	font (dialog)		

NOTE: To edit a property, the use of a resource type editor is recommended. Click on the *Resource Type Editor* symbol ( ) to open a local menu and open an appropriate editor. Alternatively a property may be edited directly through the *Translation Edit* bar.

3.3.4.4 Dialog Box Editor

RC-WinTrans' Dialog Editor (a.k.a. "layout edit mode") lets you change the size of a dialog box and modify the size and/or position of individual dialog box controls. Use the *Layout Editing* toolbar button (shown below) to activate/deactivate the Editor and switch from translation mode to layout edit mode and vice versa.

The *Layout Editing* toolbar button used to activate/deactivate RC-WinTrans' Dialog Box Editor (layout edit mode).

Translation Mode vs. Layout Edit Mode

When the *Dialog Box* view is in the (default) translation mode you can select the text of a control item with the mouse or using the Shift+Tab key combination. The text of the selected control item or dialog box caption appears in the *Translation Edit* bar where it can be translated/edited. As a safeguard against any unwanted/accidental changes, **the translation mode is the default mode in which only controls with text can be selected.** The layout edit mode (Dialog Box Editor) lets you select individual controls or the dialog box itself for modification.

Layout changes can be made in one of two ways:

1. After activating the Dialog Box Editor, a layout editing frame is shown around the selected control or dialog box. Click on the frame with the primary mouse button and hold it while “pulling” the frame into the desired size/position. The direction keys on the keyboard can also be used to reposition an item when it is selected in the layout edit mode.
2. With the Dialog Box Editor active, right mouse-click on the selected item and click on the *Properties* command in the local menu that appears. A *Properties* dialog box opens where the item’s settings can be modified.

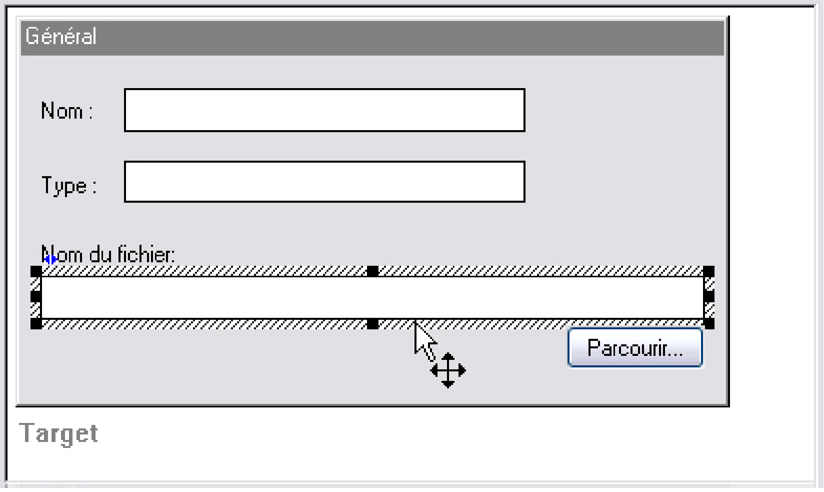


Figure: A dialog box element selected for repositioning in the *Dialog Box* view’s layout edit mode (the Dialog Box Editor).

TIP: To quickly locate dialog boxes that have undergone layout modifications use the following command to mark them in the *Project Data and Project Overview* tabs:

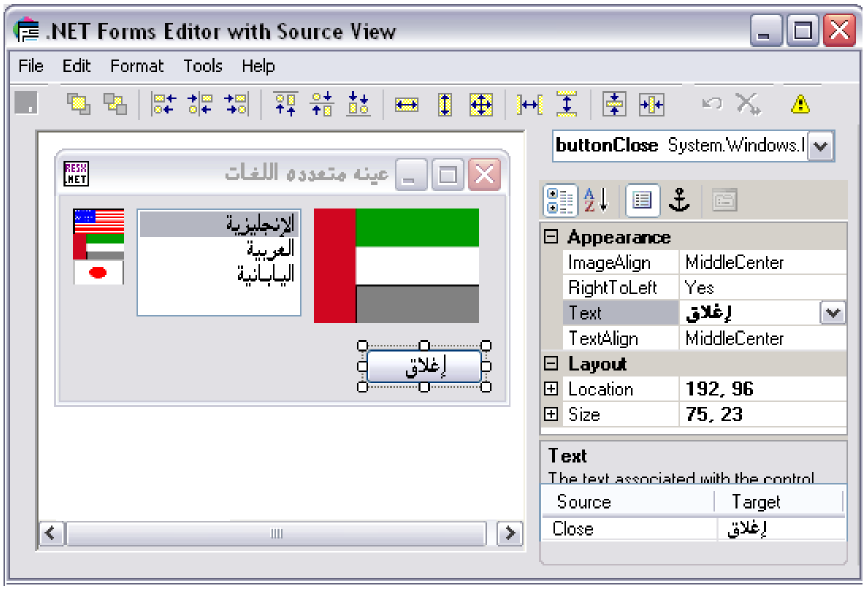
Edit menu | *Set Temporary Marker* | *Dialog Boxes with Layout Changes*

3.3.5 .NET Forms Editor

RC-WinTrans supports layout changes in Windows .NET Forms with its specialized .NET Forms Editor. The Editor can also be opened in a separate window referred to as the External (.NET Forms) Editor. The display of custom controls, Form and controls inheritance, as well as the use of .NET properties editors are all supported by the .NET Forms Editor.

The Forms Editor has a WYSIWYG (**W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et) display preview of a selected .NET Windows Form and commands for editing the Form's properties and elements. It's also equipped with a rich .NET design time support with special features such as editing user-defined types and using user-defined editors.

The Editor is equipped to handle nearly all types of .NET Windows Forms controls (standard .NET Forms controls and (third-party) custom controls). Its ability to render nearly every type of control makes it possible to translate while simultaneously controlling the appearance of the translations in context. This is not only convenient, but smart and efficient as well.



buttonClose System.Windows.I

Appearance

ImageAlign	MiddleCenter
RightToLeft	Yes
Text	إغلاق
TextAlign	MiddleCenter

Layout

Location	192, 96
Size	75, 23

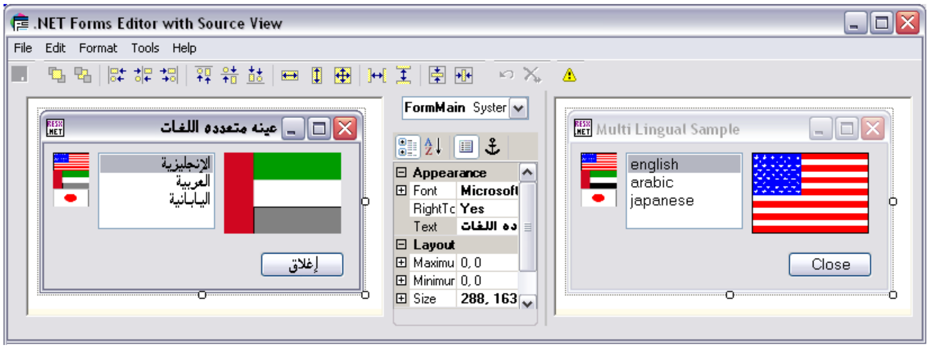
Text

The text associated with the control

Source	Target
Close	إغلاق

Figure 1: The External .NET Forms Editor with a .NET Form loaded for editing.

Figure 2: The External .NET Forms Editor with the target (left) and source (right) display of a selected Form.



Font	Microsoft
RightTc	Yes
Text	ده اللغات
Maximu	0, 0
Minimur	0, 0
Size	288, 163

NOTE: There are two variations of the .NET Forms Editor: one which is integrated into the main RC-WinTrans application and one which runs outside of it in a separate window, i.e., the External Editor.

Opening the External Editor

There are two ways to open the External Editor:

1. Use the *Open External .NET Forms Editor* toolbar button (see image that follows) in the *Project* window. The button is active when a .NET Form resource is selected in the *Text* view.
2. From the integrated *Dialog Box* view, select the *Open External .NET Forms Editor* toolbar button (shown below).

The *Open External .NET Forms Editor* toolbar button.

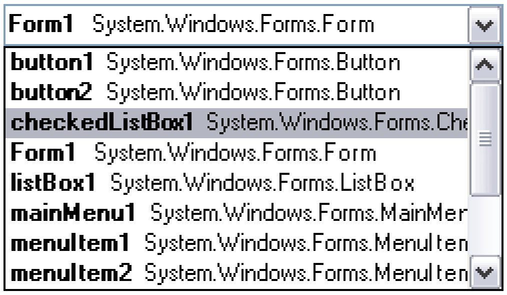
Display Area

The Editor's display area provides a design-time preview of the target (translated) Form and the source (original) Form (Figure 2). Forms and Form elements can be selected, moved, and resized in the target view (Figure 2, left side) using the keyboard keys or the mouse.

NOTE: Custom controls can only be rendered if the component (the file, DLL) containing the control is available. The Editor automatically searches for the implementation file in the same folder in which the source file (to be translated) is located. Rendering of a custom control fails if the Editor cannot find or create a custom control; a placeholder consisting of a panel surrounded by a dashed line will be displayed instead.

Component Combo-Box

The *Component* combo-box lists all the components contained in the Form. Each entry is composed of a control's name followed by its type value. The combo-box is particularly useful if a control cannot be seen (as when hidden by another control) because it forces the selection of the control, bringing it into the foreground.



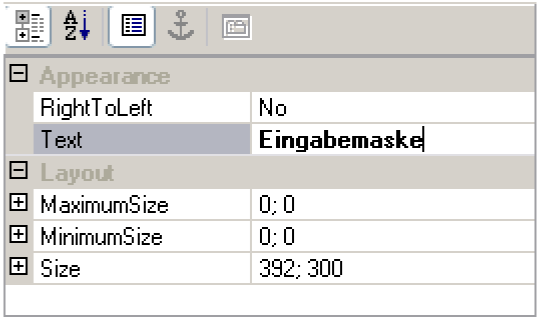
Form1 System.Windows.Forms.Form

button1 System.Windows.Forms.Button
button2 System.Windows.Forms.Button
checkedListBox1 System.Windows.Forms.Che
Form1 System.Windows.Forms.Form
listBox1 System.Windows.Forms.ListBox
mainMenu1 System.Windows.Forms.MainMer
menuItem1 System.Windows.Forms.MenuIten
menuItem2 System.Windows.Forms.MenuIten

Figure 3: The *Component* combo-box.

Property Grid

The property grid displays the properties of the currently selected control or of the propeties of the Form itself. The properties can be edited in the grid.



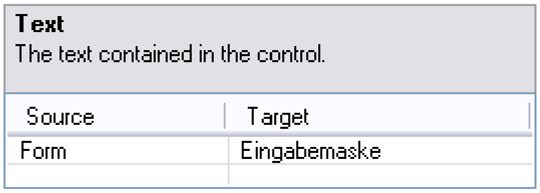
⊟ Appearance	
RightToLeft	No
Text	**Eingabemaske**
⊟ Layout	
⊞ MaximumSize	0; 0
⊞ MinimumSize	0; 0
⊞ Size	392; 300

Figure 4: The property grid in the External .NET Forms Editor.

NOTE ON THE "INHERIT" PROPERTY VALUE: If a property's value (e.g., font, backspace color, right-to-left) is set to "inherit," the property grid will display the parent's concrete value instead of the object value "inherit." This is remembered internally however, and the inherited value is used. If the Editor is closed, the term "inherit" will be saved, not the parent's concrete value. This is simply the default .NET behavior.

Source and Target List

The source and target values of a selected property are displayed in the “Source” and “Target” columns.



Text

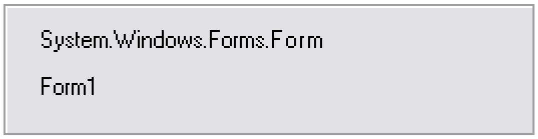
The text contained in the control.

Source	Target
Form	Eingabemaske

Figure 5: The source and target list displaying an English source text “Form” and its German translation “Eingabemaske.”

Control Type Panel

The control type panel displays the selected control’s name and type. An error message is displayed if a control cannot be displayed.



System.Windows.Forms.Form

Form1

Figure 6: The External Editor’s control type panel.

.NET Windows Forms Control Types

- STANDARD CONTROLS (.NET FRAMEWORK)
  All types of ready-to-use controls included with the .NET Framework.

- CUSTOM CONTROLS (USER CONTROLS)
  All types of controls based on the .NET Framework “Control” class, but not contained within the .NET Framework (not a standard control). There are several different types of custom controls:

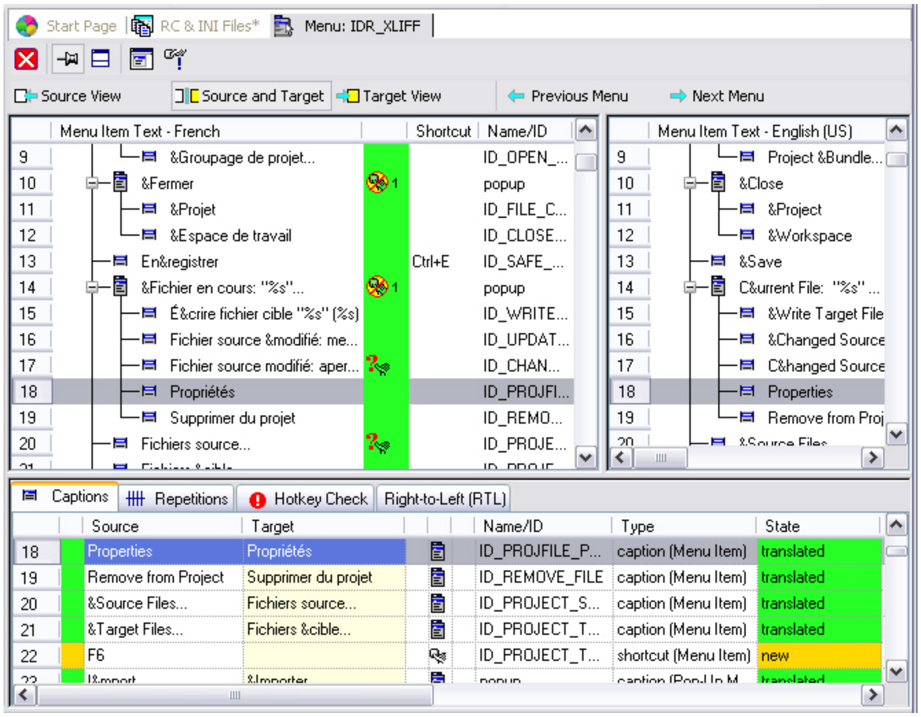
  - *COMPOSITE CONTROLS (USER CONTROLS)*
    Controls made by combining existing controls. Composite controls encapsulate a user interface that can be reused as a control.

  - *EXTENDED CONTROLS*
    Controls made by customizing any Windows Forms control by overriding or adding properties, methods, and/or events.

  - *OTHER CONTROLS*
    Controls that do not combine or extend existing controls.

3.3.6 *Menu* View

The *Menu* view shows a single menu resource as a tree view for both the source and target languages. The text of a menu item can be selected and edited directly in the view or in the *Translation Edit* bar. Additional features and commands for working with menu resources including automatic detection of missing/doubled hotkeys and a preview of translated menus are also provided.



	Menu Item Text - French		Shortcut	Name/ID
9	&Groupage de projet...			ID_OPEN_...
10	&Fermer	1		popup
11	&Projet			ID_FILE_C...
12	&Espace de travail			ID_CLOSE...
13	En&registrer		Ctrl+E	ID_SAFE_...
14	&Fichier en cours: "%s"...	1		popup
15	É&crire fichier cible "%s" (%s)			ID_WRITE...
16	Fichier source &modifié: me...			ID_UPDAT...
17	Fichier source modifié: aper...			ID_CHAN...
18	Propriétés			ID_PROJFI...
19	Supprimer du projet			ID_REMO...
20	Fichiers source...			ID_PROJE...

	Menu Item Text - English (US)
9	Project &Bundle...
10	&Close
11	&Project
12	&Workspace
13	&Save
14	C&urrent File: "%s" ...
15	&Write Target File
16	&Changed Source
17	C&hanged Source
18	Properties
19	Remove from Proj
20	&Source Files

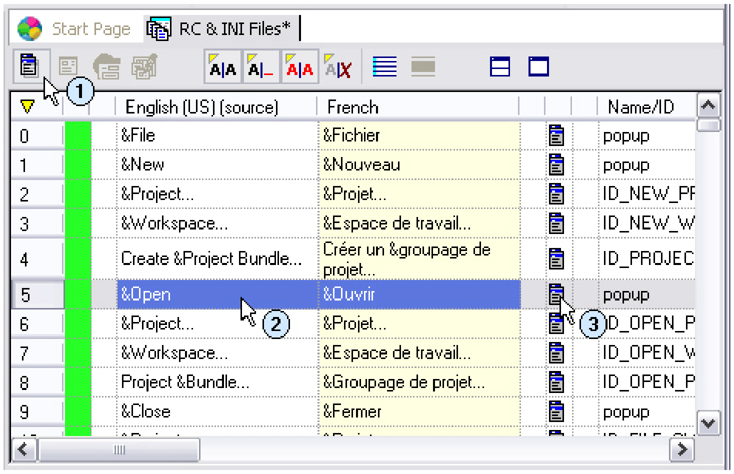
		Source	Target		Name/ID	Type	State
18		Properties	Propriétés		ID_PROJFILE_P...	caption (Menu Item)	translated
19		Remove from Project	Supprimer du projet		ID_REMOVE_FILE	caption (Menu Item)	translated
20		&Source Files...	Fichiers source...		ID_PROJECT_S...	caption (Menu Item)	translated
21		&Target Files...	Fichiers &cible...		ID_PROJECT_T...	caption (Menu Item)	translated
22		F6			ID_PROJECT_T...	shortcut (Menu Item)	new

Figure: The *Menu* view with two toolbars at the top and two menu tree views below, one for the target menu (left), and one for the source menu (right). The lower half of the view contains four tab views displaying information about the currently selected menu item (the *Captions* view is active in this image).

3.3.6.1 Opening the *Menu* View

The *Menu* view can be opened from the main window of a translation project (*Project* window) in one of three ways:

1. Click on the *Open Menu View* toolbar button.
2. Double-click on the entry in the *Text Table* view.
3. Click on the *Menu* symbol in the *Text Table* view.



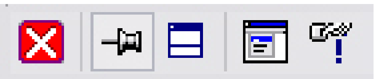
	English (US) (source)	French	Name/ID
0	&File	&Fichier	popup
1	&New	&Nouveau	popup
2	&Project...	&Projet...	ID_NEW_PF
3	&Workspace...	&Espace de travail...	ID_NEW_W
4	Create &Project Bundle...	Créer un &groupage de projet...	ID_PROJEC
5	&Open	&Ouvrir	popup
6	&Project...	&Projet...	ID_OPEN_P
7	&Workspace...	&Espace de travail...	ID_OPEN_W
8	Project &Bundle...	&Groupage de projet...	ID_OPEN_P
9	&Close	&Fermer	popup

Figure: The three ways of opening the *Menu* view as previously described.

3.3.6.2 Toolbars

The *Menu* view has two toolbars with shortcuts to commands related to the translation and display. Brief descriptions of the toolbar buttons follow.

***Menu View* Toolbar**



***Close Window* Button**
Closes the *Menu* view.

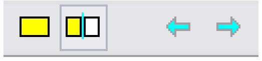
***Fixed Menu/Dialog Box View* Button**
Fixes, or designates, the view as the one view for displaying menu and dialog box resources. When activated, this view alone will be used to display menu and dialog box resources. If deactivated, a new, separate view will be opened for each menu or dialog box you want to display.

***Tile Window* Button**
Tiles the window vertically with the WYSIWYG (**W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et) display on top and the tab views at the bottom.

***Menu Preview* Button**
Opens the menu preview display where the selected menu is shown as it appears in the translated target file.

***Hotkey Suggestion* Button**
Opens the *Hotkey Suggestion* dialog box for assistance in selecting a suitable hotkey for the selected menu text. See Section 3.5.1.3, "Hotkey/Shortcut Key Check" for further details.

***Navigation* Toolbar**



***Target View* Button**
Shows the currently selected menu in tree view format. The target menu display will be hidden.

***Source and Target* Button**
Displays the source and target menus side by side.

***Previous Menu* Button**
Selects and displays the previous menu (chronologically) in the project.

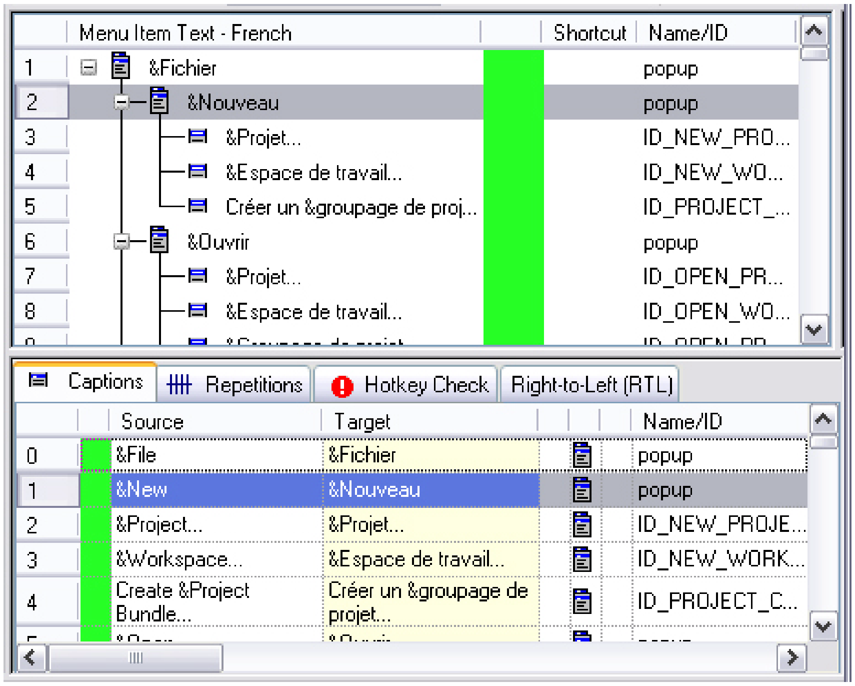
***Next Menu* Button**
Selects and displays the next menu (chronologically) in the project.

3.3.6.3 Tab Views

Like the other *Project* window views, the *Menu* view has a secondary window with different tab views to support the localization of these resource types, with each one handling a different aspect to ensure that they are localized correctly.

3.3.6.3.1 *Captions* Tab View

The *Captions* tab (see figure) lists the caption texts of all the menu items in the currently selected menu. As with the *Text Table* view, an item listed in the *Captions* tab view can be selected and translated in the *Translation Edit* bar, or it can be edited directly in the table (inline editing).



	Menu Item Text - French		Shortcut	Name/ID
1	&Fichier	🟩		popup
2	&Nouveau	🟩		popup
3	&Projet...	🟩		ID_NEW_PRO...
4	&Espace de travail...	🟩		ID_NEW_WO...
5	Créer un &groupage de proj...	🟩		ID_PROJECT_...
6	&Ouvrir	🟩		popup
7	&Projet...	🟩		ID_OPEN_PR...
8	&Espace de travail...	🟩		ID_OPEN_WO...
9	&Groupage de projet	🟩		ID_OPEN_PR

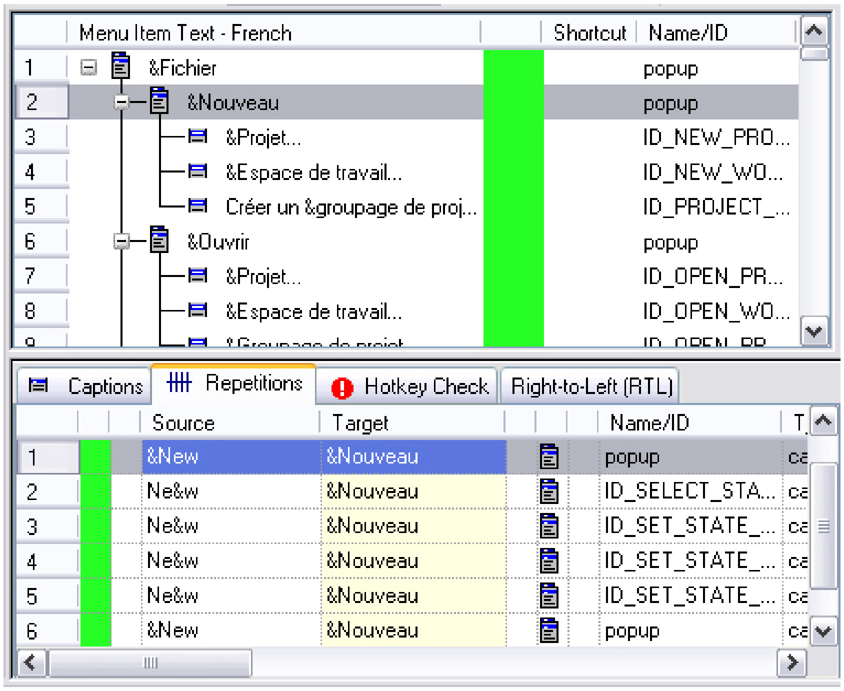
		Source	Target	Name/ID
0	🟩	&File	&Fichier	popup
1	🟩	&New	&Nouveau	popup
2	🟩	&Project...	&Projet...	ID_NEW_PROJE...
3	🟩	&Workspace...	&Espace de travail...	ID_NEW_WORK...
4	🟩	Create &Project Bundle...	Créer un &groupage de projet...	ID_PROJECT_C...
5	🟩	&Open	&Ouvrir	popup

Figure: The *Menu* view with the menu tree view on the top and the *Captions* tab view active below it. The *Captions* tab lists all the texts from the currently selected menu.

3.3.6.3.2 *Repetitons* Tab View

The *Repetitions* tab view (see figure) lists all occurrences of texts which are identical to the text of the item currently selected in the *Menu* view. All translation units that have the same source text are considered repetitions. The view supports inline editing just like the *Text Table* view does.

See also: Section 3.5.8, "Translating Repetitions."



	Menu Item Text - French		Shortcut	Name/ID
1	&Fichier	🟩		popup
2	&Nouveau	🟩		popup
3	&Projet...	🟩		ID_NEW_PRO...
4	&Espace de travail...	🟩		ID_NEW_WO...
5	Créer un &groupage de proj...	🟩		ID_PROJECT_...
6	&Ouvrir	🟩		popup
7	&Projet...	🟩		ID_OPEN_PR...
8	&Espace de travail...	🟩		ID_OPEN_WO...
9	&Groupage de projet	🟩		ID_OPEN_PR

		Source	Target		Name/ID	T
1	🟩	&New	&Nouveau		popup	ca
2	🟩	Ne&w	&Nouveau		ID_SELECT_STA...	ca
3	🟩	Ne&w	&Nouveau		ID_SET_STATE_...	ca
4	🟩	Ne&w	&Nouveau		ID_SET_STATE_...	ca
5	🟩	Ne&w	&Nouveau		ID_SET_STATE_...	ca
6	🟩	&New	&Nouveau		popup	ca

Figure: The *Menu* view with the *Repetitions* tab showing multiple repetitions for the selected source text ("&New").

3.3.6.3.3 *Hotkey Check* Tab View

Doubled and missing hotkeys (mnemonic keys) for menu resources are automatically listed in the *Menu* view's *Hotkey Check* tab view- a check does not need to be run manually.

The *Hotkey Check* tab lists all caption texts with missing or doubled hotkeys. The hotkey check and listing of hotkey errors are performed automatically- there is no need to run a check manually. The absence/presence of hotkey errors is indicated by one of two different symbols next to the tab title.

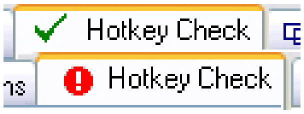


Figure: The tab of the *Hotkey Validation* tab view. A green check mark indicates that the menu contains no hotkey conflicts (no missing/doubled hotkeys). A red circle with exclamation mark (*Alert* symbol) indicates that the menu contains hotkey conflicts needing to be resolved. The affected control items are listed in the view.

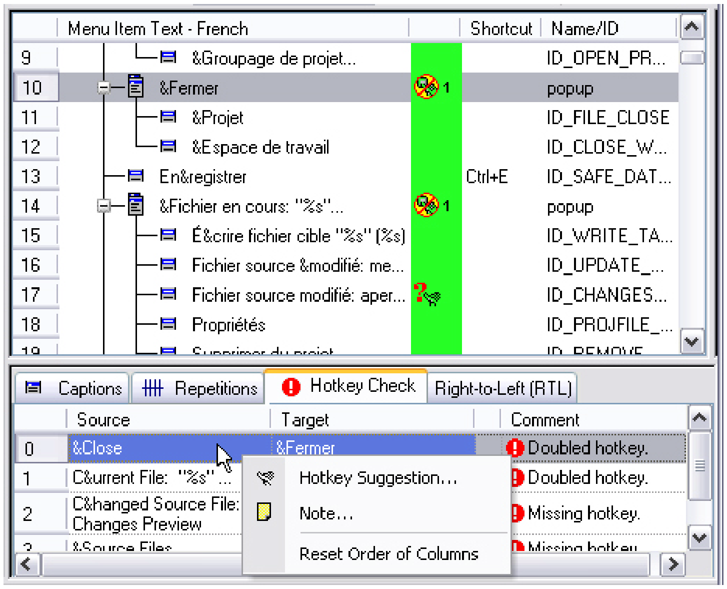
Doubled and missing hotkeys are immediately indicated in the menu tree view by the following symbols:

Missing Hotkey
This symbol indicates that the source text for the item contains a hotkey but the translation text does not.

1
1

Doubled Hotkeys
The appearance of this symbol indicates that the same hotkey has been assigned to two or more menu items. The number shown next to the symbol shows which items share the same hotkey.

Hotkey errors and a discussion of how to resolve them can be found in Section 3.5.1.3 of this guide ("Hotkey/Shortcut Key Check").



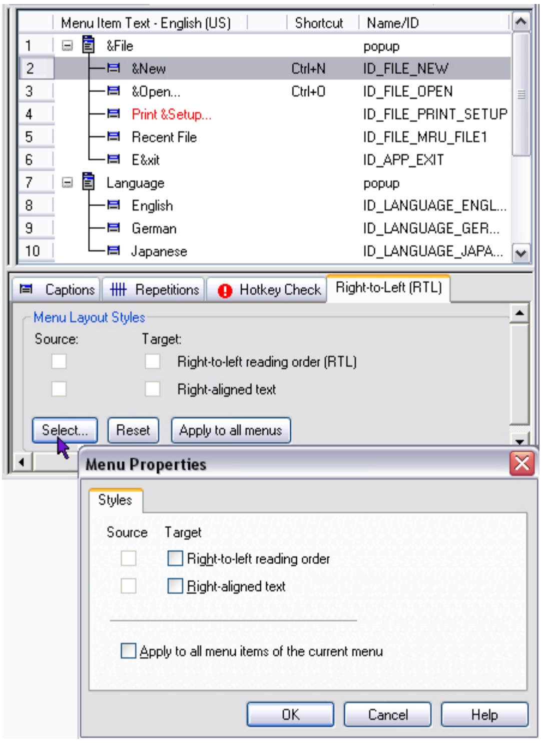
	Menu Item Text - French		Shortcut	Name/ID
9	&Groupage de projet...			ID_OPEN_PR...
10	&Fermer	⊘ 1		popup
11	&Projet			ID_FILE_CLOSE
12	&Espace de travail			ID_CLOSE_W...
13	En&registrer		Ctrl+E	ID_SAFE_DAT...
14	&Fichier en cours: "%s"...	⊘ 1		popup
15	É&crire fichier cible "%s" (%s)			ID_WRITE_TA...
16	Fichier source &modifié: me...			ID_UPDATE_...
17	Fichier source modifié: aper...	?		ID_CHANGES...
18	Propriétés			ID_PROJFILE_...

	Source	Target	Comment
0	&Close	&Fermer	Doubled hotkey.
1	C&urrent File: "%s" ...		Doubled hotkey.
2	C&hanged Source File: Changes Preview		Missing hotkey.

Figure: The *Menu* view with hotkey errors indicated in the menu tree view at the top and the *Hotkey Check* tab view active below it. Here the menu item in line 17 is missing a hotkey while a doubled hotkey ("&F") exists for the two menu items in lines 10 and 14. A local menu has been opened to access the *Hotkey Suggestion* dialog box (described in Section 3.5.1.3).

3.3.6.3.4 *Right-to-Left (RTL)* Tab View

The *Right-to-Left (RTL)* tab is used to set two menu layout style options, right-aligned text and right-to-left reading order. Press the *Select* button to select/unselect these options for the selected menu item alone or for all the items in the menu. The selections can be applied to all the menus in the translation project by clicking on the *Apply to all menus* command button.



	Menu Item Text - English (US)	Shortcut	Name/ID
1	&File		popup
2	&New	Ctrl+N	ID_FILE_NEW
3	&Open...	Ctrl+O	ID_FILE_OPEN
4	Print &Setup...		ID_FILE_PRINT_SETUP
5	Recent File		ID_FILE_MRU_FILE1
6	E&xit		ID_APP_EXIT
7	Language		popup
8	English		ID_LANGUAGE_ENGL...
9	German		ID_LANGUAGE_GER...
10	Japanese		ID_LANGUAGE_JAPA...

Figure: The *Menu* view with the *Right-to-Left (RTL)* tab view active at the bottom.

The *Menu Properties* dialog box has been opened to set the style of the menu/menu item.

3.3.7 Localizing Images

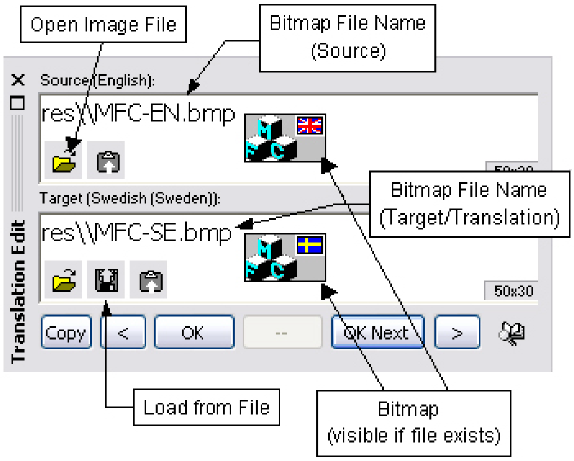
RC-WinTrans lets you “translate” images, or use a different image for the target file than the one used in the source file. For Win32 resource files, images are usually Windows bitmaps. For .NET resource files, images can be any image format (e.g., Windows bitmap (BMP), JPEG, GIF, or TIFF).

Image resources are listed in *Project* window (*Text Table* view), and image “translation” (modification) is done through the *Translation Edit* bar. The meaning of “image translation” depends on whether the source file is a Win32 resource script file (RC), a program file (EXE, DLL, or OCX), or a .NET Windows Forms file. In general, images are created and/or edited with an external graphics tool (Figure 5) and RC-WinTrans takes the modified image and uses it for the target file.

Win32: Images in Resource Script (RC) Files

Bitmaps (binary data) are not actually part of a resource file, but are stored separately in single bitmap files. The RC file contains the name of the bitmap file, where the file might be put together with a relative path specification plus the file name. In this context, “translating” bitmaps means (1) having or creating a bitmap file other than the one used in the original source file; and (2) entering this file name in RC-WinTrans’ *Translation Edit* bar as the translation for the original file name.

NOTE: In principle, bitmaps can be included in a resource script file as binary data. RC-WinTrans **does not** support binary bitmap translation in resource script files.



Source (English): res\\MFC-EN.bmp

Target (Swedish (Sweden)): res\\MFC-SE.bmp

Open Image File
Bitmap File Name (Source)
Bitmap File Name (Target/Translation)
Translation Edit
50x30
Copy
<
OK
--
OK Next
>
Load from File
Bitmap (visible if file exists)

Figure 1: The *Translation Edit* bar. The *Source* edit field shows the original bit-map file name (“res\\MFC-EN.bmp”) in addition to the bitmap itself. The file name of the target bitmap has been entered in the *Translation* edit field (“res\\MFC-SE.bmp”).

Images in Win32 RC files are “translated” as follows:

1. Select the image in the *Text Table* view.
2. In the *Translation Edit* bar (Figure 1), use the command below the *Source* edit field to open the source image. RC-WinTrans will open the image file with an external graphics tool (Figure 3).
3. Save the opened source file under a different file name (the target file name) with the graphics tool.
4. Edit the image with the external graphics tool and save the file afterwards.
5. Use the *Load from File* command below the *Target* edit field in the *Translation Edit* bar and select the image file. RC-WinTrans will use the name of the file as the file name in the translated RC file.

Images in Win32 EXE/DLL Files and .NET Files

Win32 binary files and all .NET files have images included as binary data in the source file. An image is "translated" by editing the image file externally (with an external tool) and using the *Load from File* command (*Translation Edit* bar) to tell RC-WinTrans to copy the data from the file into the translation database. The binary data will be included in the translated target file.

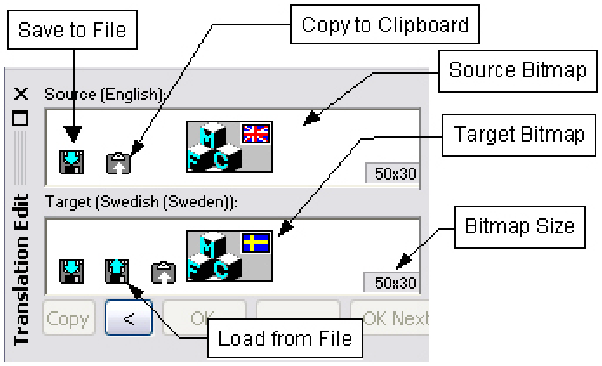


Figure 2: The *Translation Edit* bar. The *Source* edit field shows the original (source) image and the *Target* edit field shows the “translated” (target) image.

Images in Win32 EXE/DLL and .NET files are “translated” as follows:

1. Select the image in the *Text Table* view.
2. In the *Translation Edit* bar, use the *Save to File* command below the *Source* edit field (Figure 2) to save the source image to a file for editing.
3. Edit the image with the external graphics tool and save the file afterwards.
4. Use the *Load from File* command below the *Target* edit field in the *Translation Edit* bar and select the image file. RC-WinTrans will import the image data from the file to be used as the target image.

The *Copy to Clipboard* command can be used to copy the image to the clipboard and paste it into an external graphics application where it can be modified and saved as the target image in the manner described above.

External Graphics Tools

The external graphics tool used to modify specific image types can be specified on the *External Tools* tab in the *Options* dialog box. RC-WinTrans uses Microsoft Paint ("mspaint.exe") as the default application for opening and editing images.

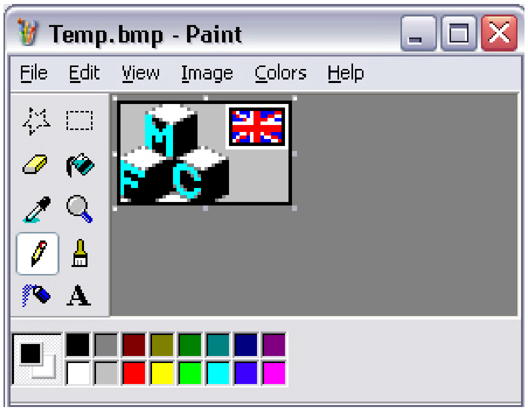


Figure 3: The bitmap from the source file shown in Figures 1 and 2 opened and ready to be edited in Microsoft Paint.

Deleting an Image

To delete a target image, select the image in the *Text Table* view and use the "Delete" key or the *Delete* command button in the *Translation Edit* bar.

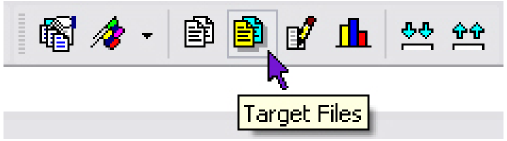
Check Function for Images

RC-WinTrans is equipped with a special check function to verify that a translated image has the same size and color depth as the original image.

3.3.8 Writing Target Files

Target files are written from the *Target Files* tab page in the *Source and Target Files* dialog box. Open the dialog box by clicking on the *Target Files* button in the *Project* toolbar (see figure below), or by selecting the *Target Files* command from the *File* or the *Project* menu.

Figure: The *Target Files* button in the *Project* toolbar.



A target file can be written at any time and with any translation status (translated or not). A target language does not need to be translated/localized in order to write a target file. If a text item is not translated the source text will always be used. This applies to other data such as bitmaps or accelerator keys (shortcuts) as well: when no target data exists for an item, the source data will be used (contained in) the target file.

Excluded Text
When an item is excluded from translation, the source text will always be used in the target language file. There is both a toolbar button (*Item State* toolbar) and a menu command (under *Attributes* in the *Edit* menu) for excluding items.

Pseudo-Translation
Pseudo-translation simulates the effects of special characters and decreased or increased string lengths on the software to verify that it will continue to work properly in the target language.

Read more about pseudo-translation in Section 3.3.8.2 of this guide.

Resource Script Files (Win32 RC Files)
RC-WinTrans lets you write single-language target files when localizing resource script (RC) files with commands available on the *Target Files* page (*Source and Target Files* dialog box).

RC-WinTrans **does not** change the format of a target file when it is written. The target file is an identical copy of the original source file where the language-dependent data is replaced by the localized data. It is therefore essentially a copy of the source file. A target file can **only** be written if the source file is available.

Executable Files (Win32 EXE, DLL, OCX)
Single-language program (executable) files can be written using the commands available on the *Target Files* tab (*Source and Target Files* dialog).

Use the *Target Files (Multiple-Language)* tab (same dialog box) to write a multiple-language target file (Win32 EXE and DLL files only).

RC-WinTrans does not write a target file from scratch; instead, it copies the source file and uses Windows API functions afterwards to manipulate, add, or delete resources in the target file. This causes the target file to have a different size but it does not affect the use of the program file whatsoever.

The difference in file size between the source and target files is dependent upon the technique used to write the target file: if RC-WinTrans only changes the data of existing resources, the target file will be about the same size as the source file. However, if RC-WinTrans has to change the language ID of the resource, the original resources must be deleted and the resources for the target language must be added as new resources. As a result, the target file becomes fragmented during the process and ultimately becomes bigger than the source file. The increase in size depends on the number of resources. Again, the fragmentation of the target file and the difference in size **does not** affect the use of the program file in any way.

The *Set Language ID* option can be checked or unchecked on the *Target Files* tab (*Source and Target Files* dialog).

.NET Files

RC-WinTrans writes .NET target files according to source file type (RESX, RESOURCES, or TXT). For binary .NET executable (EXE) and component (DLL) files, RC-WinTrans creates satellite DLLs, one for each target language.

Conceptually, a localized .NET file contains (is written with) the localized data only. Thus a localized target file only contains the “data difference,” or the data that is different from the data contained in the source file.

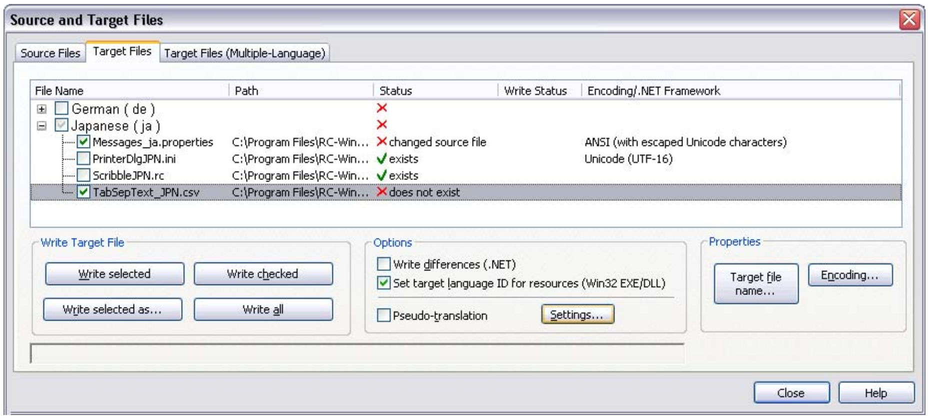
Alternatively, a target file can be created as a “full” file containing all localized data as well as source data with no translation (making the target texts for such data identical to those in the source). This option (“full mode”) can be specified on the *Target Files* tab (*Source and Target Files* dialog box) by **un**checking the *Write differences (.NET)* option found there.

Satellite DLLs

For binary .NET executable (EXE) and component (DLL) files, RC-WinTrans creates satellite DLLs, one for each target language. By default, a satellite DLL is written to the subdirectory where the source file is located. The two letters of the target language’s country code are used for the name of the subdirectory.

3.3.8.1 *Target Files* Tab Page

The *Target Files* tab page in the *Source and Target Files* dialog box is used to write (translated) target files and to set the encoding of a target file. Here you will find a list of the files contained in the translation project along with an indication of their current status (target file exists or target file not up-to-date). The *Pseudo-translation* option lets you write target files with a pseudo-translation text (see Section 3.3.8.2, "Pseudo-Translation").



File Name	Path	Status	Write Status	Encoding/.NET Framework
☐ German (de)		✗		
☑ Japanese (ja)		✗		
☑ Messages_ja.properties	C:\Program Files\RC-Win...	✗ changed source file		ANSI (with escaped Unicode characters)
☐ PrinterDlgJPN.ini	C:\Program Files\RC-Win...	✓ exists		Unicode (UTF-16)
☐ ScribbleJPN.rc	C:\Program Files\RC-Win...	✓ exists		
☑ TabSepText_JPN.csv	C:\Program Files\RC-Win...	✗ does not exist		

☐ Write differences (.NET)
☑ Set target language ID for resources (Win32 EXE/DLL)
☐ Pseudo-translation

Figure: The *Source and Target Files* dialog box with the *Target Files* tab page active.

Symbols

✓ A green check mark in the "Status" column indicates that a target file exists and that it is up-to-date, **OR** that a command has been performed successfully.

✗ A red "X" mark indicates that a target file does not exist or that it is not up-to-date, **OR** that a command has not been performed successfully.

File Status

There are several different file statuses that can appear in the "Status" and "Write Status" columns of the *Target Files* listbox. Please refer to the table below for their meanings.

STATUS	DESCRIPTION
changed source file	The corresponding source file has changed. The translation database must be updated before the target file can be written.
does not exist	The target file does not exist.
exists	The target file exists and is up-to-date.
not up-to-date	The translation data has been modified since the target file was last created. The target file must be rewritten.
read-only	The target file is write-protected and cannot be over-written by RC-WinTrans.

Commands and Options

Write selected Writes the (single) selected target file.

Write selected as Opens a "Save File" dialog box to specify the name and location to be used to write the target file.

Write checked Writes all the target files checked in the *Target Files* listbox.

Write all Writes all the target files contained in the project.

Write differences (.NET) **.NET files only.** Check this option to write only the data that has been localized to the target file, i.e., data that is different from the data contained in the source file. If this option is unchecked the target file will be written with the localized data plus the source data for non-localized entries.

Pseudo-translation Check this option to write the target file(s) using pseudo-translation text.

Settings Opens the *Pseudo-Translation Properties* dialog box.

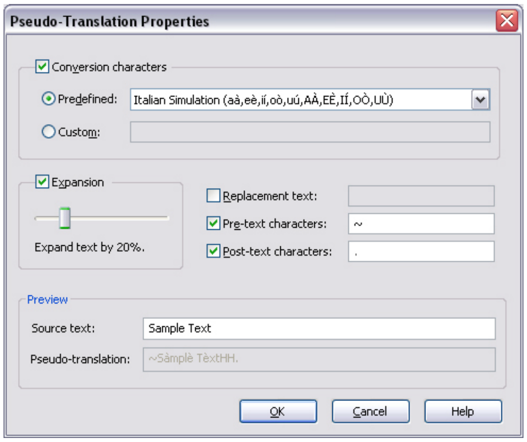
Target file name Opens the *Target File Names* dialog box where the target file path and name can be specified.

Encoding Opens the *Target File Encoding* dialog box for the (single) selected target file.

3.3.8.2 Pseudo-Translation

Pseudo-translation simulates the effects of special characters and decreased or increased string lengths on the software to verify that it will continue to work properly in the target language.

Enabling the pseudo-translation option replaces all the text items (in the target file) with a pseudo-translation text specified in the *Pseudo-translation Properties* dialog box. Open this dialog box from the *Target Files* page (*Source and Target Files* dialog box) by pressing the *Settings* button next to the *Pseudo-translation* checkbox.



Pseudo-Translation Properties

☑ Conversion characters

● Predefined: Italian Simulation (aà,eè,ií,oò,uú,AÀ,EÈ,IÍ,OÒ,UÙ)

○ Custom:

☑ Expansion

Expand text by 20%.

□ Replacement text:

☑ Pre-text characters: ~

☑ Post-text characters: .

Preview

Source text: Sample Text

Pseudo-translation: ~Sàmplè TèxtHH.

Figure: The *Pseudo-Translation Properties* dialog box.

3.3.8.3 *Target File Names* Property Page

The *Target File Names* property page lets you define a rule specifying a target path and a target file name to be used for all of the project's target files (of a certain type) or for the current target file only. It is opened by clicking on the *Target file names* button found at the bottom of the *Target Files* tab.

The path and name of a target file may depend on your specific project and must usually follow the rules of the software development platform with which the software project (the source files) were created. For example, the target file path and name of a .NET assembly DLL follows (according to the development platform) different rules than the properties files of a Sun Java project.

Example (Chinese (Traditional) target language):

The target file path and name of a.NET assembly DLL follow (according to the development platform) different rules than the properties files of a Sun Java project:

.NET – Source file: “C:\MyProject\MyAppl.exe”
Target file: “C:\MyProject**\zh-tw**\MyAppl.resources.dll”

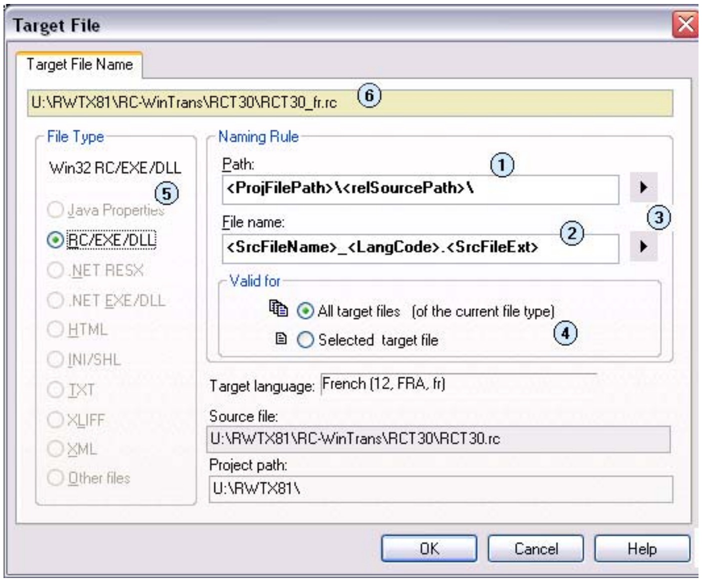
Java – Source file: “C:\MyProject\MyAppl.properties”
Target file: “C:\MyProject\MyAppl**_zh_TW**.properties”

As shown in the example above, a .NET target file (assembly DLL) is written to a subdirectory of the executable file where the name of the subdirectory is the locale language code (string) of the target language. The target file of a Java properties file is usually written to the same directory as the source file, and the name of a Java properties file follows the rule that the target file name is a combination of the name of the source file plus the locale language string of the target language.

Since a target file is created using specific rules for the path and file name, these rules must be made definable. This can be done on the *Target File Names* property page.

NOTE – “Rules per File Type”: By default, a target file rule implemented in RC-WinTrans will be defined for all the files of that particular type within a translation project. The rule can also be specified for a single target file.

Figure: The *Target File Names* property page showing the default rules for RC, EXE and DLL files.



Target File

Target File Name

U:\RWTX81\RC-WinTrans\RCT30\RCT30_fr.rc

File Type

Win32 RC/EXE/DLL

○ Java Properties
● RC/EXE/DLL
○ .NET RESX
○ .NET EXE/DLL
○ HTML
○ INI/SHL
○ TXT
○ XLIFF
○ XML
○ Other files

Naming Rule

Path: <ProjFilePath>\<relSourcePath>\

File name: <SrcFileName>_<LangCode>.<SrcFileExt>

Valid for

● All target files (of the current file type)
○ Selected target file

Target language: French (12, FRA, fr)

Source file: U:\RWTX81\RC-WinTrans\RCT30\RCT30.rc

Project path: U:\RWTX81\

① The naming rule for the target file path.

② The naming rule for the target file name.

③ Command buttons to select a definition placeholder for the corresponding rule on the left.

④ Specification of the naming rules as valid for *all* the target files of the selected file type or for the current target file only.

⑤ File type for which the file name and path rule are valid.

⑥ Target file path and target file name based on the specified rules.

The two primary fields of the *Target File Names* property page are (1) the *Path* edit field which defines the target path rule and (2) the *File name* edit field which defines the rule for the target file name. Placeholders are used (in a rule) in both of these fields to define the target path and name. A rule can also contain statically-named path names, subfolders, and/or file names.

Select and add a placeholder by clicking on the appropriate button (#3 in the previous figure) to open a local menu with a selection of placeholders.

Placeholders for Language Encodings and Path Variables

There are four language encoding placeholders that can be used to extend the target file name or path. The following tables describe the placeholders that can be used in the path and file name rule definitions:

Language Placeholder	Description	Example
<WinLangID>	Windows language ID	“1028”
<WinLangCode>	Windows language code	“CHT”
<JavaLangCode>	Java language code	“zh_TW”
<LangCode>	Language/Region code	“zh-tw”

Path and File Name Placeholder	Description	Example
<ProjFilePath>	The path of the project definition file also named as the project path.	“C:\MyProject”
<relSourcePath>	The path to the source file relative to the project path.	“JavaFiles\”
<absSourcePath>	The source file path (absolute path).	“C: \MyProject\ JavaFiles”
<SrcFileName>	The file name of the source file without the file extension.	“MyAppl”
<SrcFileExt>	The file extension of the source file.	“properties”
<SrcFileName.Ext>	The file name plus the file extension of the source file.	“MyAppl.properties”

Calling a VBA Macro

RC-WinTrans can call a VBA macro each time the name of the target file is requested. For example, a target file name (and the path) is requested each time RC-WinTrans performs a *Write Translated Target File* operation. The VBA macro can be used to return the target file name and path to ignore or modify the defined rules.

RC-WinTrans can call a VBA macro each time the name of the target file is requested. For example, a target file name (and path) is requested each time RC-WinTrans generates a translated target file.

The VBA macro can be used to return the target file name and a target path to be used (ignoring or modifying the defined rules).

Using a VBA macro to return the target file name is useful for example if the possibility to define the rules on the *Target File Names* property page is not sufficient for the user's requirements.

Information on how to create a user macro for this purpose can be found in the RC-WinTrans Help ("Target File Name Macro").

3.4 Source File Changes Handling

3.4.1 Source File Changes Detection and Indication

Changes Detection

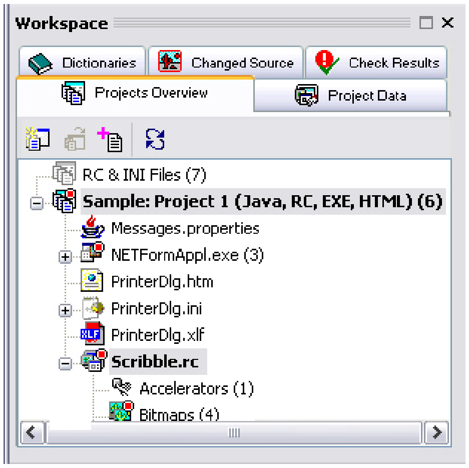
RC-WinTrans features specialized (translation database) update handling in instances where the source file has changed and the project's translation data (translation database) need to be updated so that they will contain the same (source) data. The data contained in the current source file is compared to the source data contained in the file that is being worked on in order to determine what exactly has changed. You then have the option of updating the translation data right away or of generating first a changes report that lists all the changes (additions, deletions, and/or modifications) in the source file(s). RC-WinTrans also makes sure that none of the existing translation data gets lost during the update process. New data in a source file will be added to the translation database and removed data will also removed from the translation database.

The source files of a translation project are automatically checked for changes each time a project is opened. A check for changed source files can be performed manual5ly by pressing the F5 function key. After pressing the F7 key you can check the *Projects Overview* tab view for changes indicated for source files.

Changes Indication

A changed source file is indicated in the *Projects Overview* tab (Figure 1) and on the *Source Files* tab in the *Source and Target Files* dialog box (Figure 2).

The *Projects Overview* tab view (*Workspace* bar) will show a red dot next to any changed files and next to the project to which the file(s) belong(s). As an extra help to the user, moving the mouse over a file item displays the full path and file name of the source file and the target file (of the active target language).



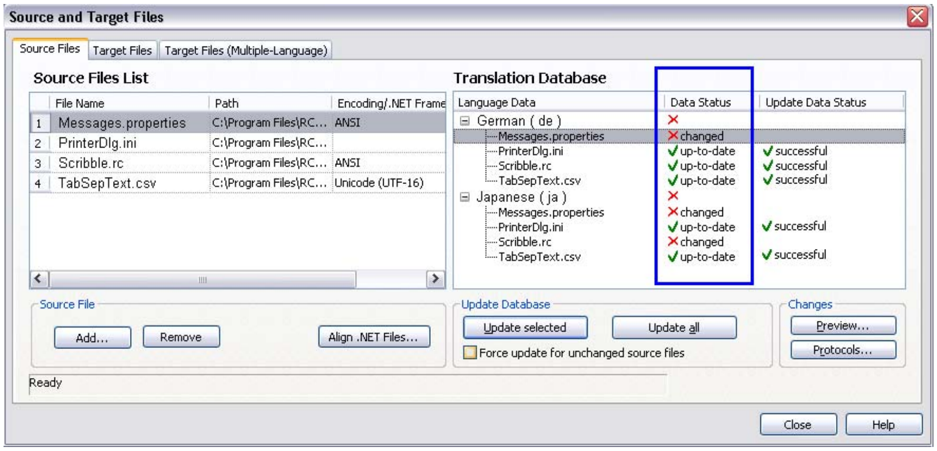
- RC & INI Files (7)
- Sample: Project 1 (Java, RC, EXE, HTML) (6)
  - Messages.properties
  - NETFormAppl.exe (3)
  - PrinterDlg.htm
  - PrinterDlg.ini
  - PrinterDlg.xlf
  - Scribble.rc
    - Accelerators (1)
    - Bitmaps (4)

Figure 1: The *Projects Overview* tab with red dot markers next to the source file "PrinterDlg.ini" and the project "Project 1 (Sample)" indicating that these have changed in the source.

Changes in the source file are indicated on the *Source Files* tab page (*Source and Target Files* dialog box). This page is opened in one of two ways:

- use the *Source Files* button in the *Project* toolbar (shown right), **or**
- use the *Source Files* command from the *Project* menu.

Changed files will be marked with a red "X" symbol and the text "changed" in the "Data Status" column (Figure 2). The commands for updating the database and for generating a preview of the changes before the update are both found on this page.



Source Files List

	File Name	Path	Encoding/.NET Frame
1	Messages.properties	C:\Program Files\RC...	ANSI
2	PrinterDlg.ini	C:\Program Files\RC...	
3	Scribble.rc	C:\Program Files\RC...	ANSI
4	TabSepText.csv	C:\Program Files\RC...	Unicode (UTF-16)

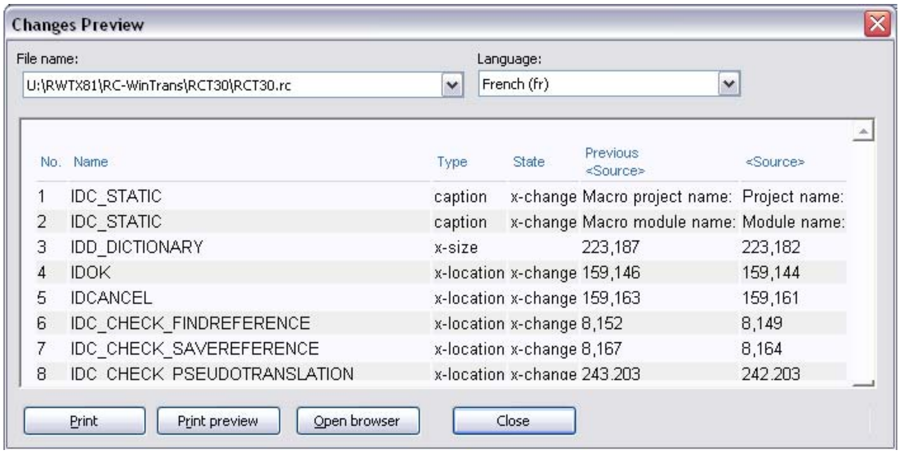
Translation Database

Language Data	Data Status	Update Data Status
German (de)	✗	
Messages.properties	✗ changed	
PrinterDlg.ini	✓ up-to-date	✓ successful
Scribble.rc	✓ up-to-date	✓ successful
TabSepText.csv	✓ up-to-date	✓ successful
Japanese (ja)	✗	
Messages.properties	✗ changed	
PrinterDlg.ini	✓ up-to-date	✓ successful
Scribble.rc	✗ changed	
TabSepText.csv	✓ up-to-date	✓ successful

Figure 2: The "Data Status" column marked on the *Target Files* tab page.

Changes Preview

RC-WinTrans can create a list of all the changes that have been made to a source file(s), providing a preview of the data that will be affected when the database update is performed. To generate the preview, select the changed files that you want to preview in the *Translation Database* display on the *Source Files* tab (Figure 2), then press the *Preview* button located below it in the *Changes* groupbox. The *Changes Preview* dialog box (Figure 3) will open with a list of the changes.



Changes Preview

File name: U:\RWTX81\RC-WinTrans\RCT30\RCT30.rc

Language: French (fr)

No.	Name	Type	State	Previous <Source>	<Source>
1	IDC_STATIC	caption	x-change	Macro project name:	Project name:
2	IDC_STATIC	caption	x-change	Macro module name:	Module name:
3	IDD_DICTIONARY	x-size		223,187	223,182
4	IDOK	x-location	x-change	159,146	159,144
5	IDCANCEL	x-location	x-change	159,163	159,161
6	IDC_CHECK_FINDREFERENCE	x-location	x-change	8,152	8,149
7	IDC_CHECK_SAVEREFERENCE	x-location	x-change	8,167	8,164
8	IDC_CHECK_PSEUDOTRANSLATION	x-location	x-change	243,203	242,203

Figure 3: The *Changes Preview* dialog box.

3.4.2 Updating the Translation Database

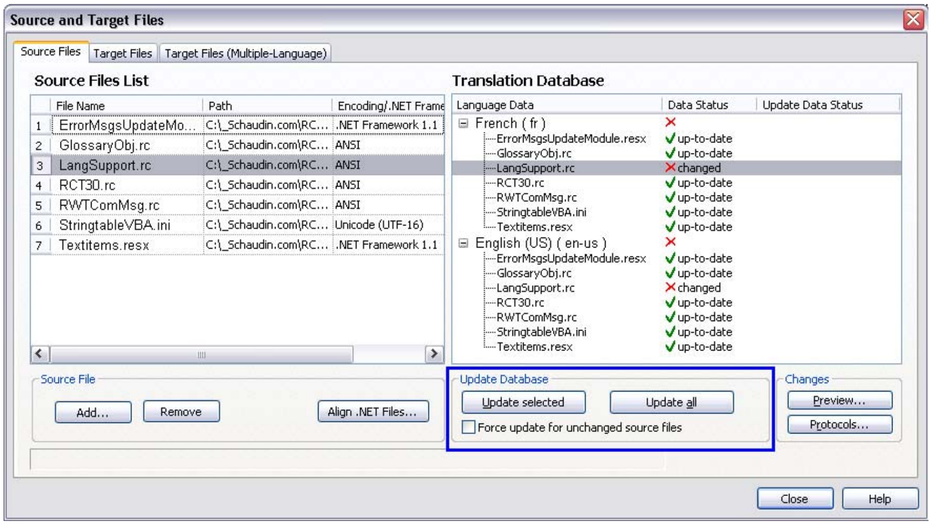
When a source file has changed the translation database must be updated to include all the latest data.

Updated data includes:

1. New items in the source file that need to be added to the translation project.
2. Existing source items for which the source data has changed must be updated in the project (translated items are marked as “changed”).
3. Items that have been removed from the source file must also be removed from the translation project.

The commands to perform an update of the project's translation database are found on the *Source Files* tab page (opened with the *Source Files* toolbar button or the *Project* | *Source Files* menu command). Select the changed files that you want to update in the *Translation Database* display, then press the *Update selected* button located below it in the *Update Database* groupbox. Use the *Update all* button to update all the files contained in the translation project.

Figure 1: The update command buttons marked on the *Source Files* tab in the *Source and Target Files* dialog box. The commands used to generate a preview of source file changes are located to the left of these.

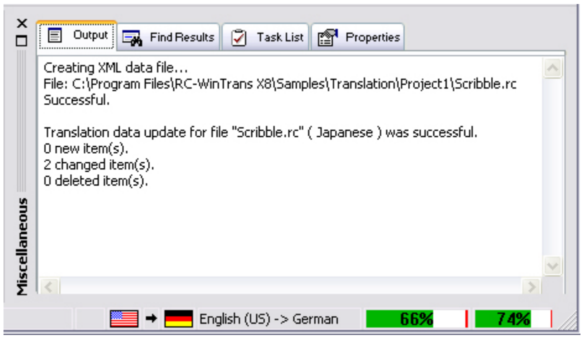


	File Name	Path	Encoding/.NET Frame
1	ErrorMsgsUpdateMo...	C:_Schaudin.com\RC...	.NET Framework 1.1
2	GlossaryObj.rc	C:_Schaudin.com\RC...	ANSI
3	LangSupport.rc	C:_Schaudin.com\RC...	ANSI
4	RCT30.rc	C:_Schaudin.com\RC...	ANSI
5	RWTComMsg.rc	C:_Schaudin.com\RC...	ANSI
6	StringtableVBA.ini	C:_Schaudin.com\RC...	Unicode (UTF-16)
7	Textitems.resx	C:_Schaudin.com\RC...	.NET Framework 1.1

Language Data	Data Status	Update Data Status
⊟ French (fr)	✗	
ErrorMsgsUpdateModule.resx	✓ up-to-date	
GlossaryObj.rc	✓ up-to-date	
LangSupport.rc	✗ changed	
RCT30.rc	✓ up-to-date	
RWTComMsg.rc	✓ up-to-date	
StringtableVBA.ini	✓ up-to-date	
Textitems.resx	✓ up-to-date	
⊟ English (US) (en-us)	✗	
ErrorMsgsUpdateModule.resx	✓ up-to-date	
GlossaryObj.rc	✓ up-to-date	
LangSupport.rc	✗ changed	
RCT30.rc	✓ up-to-date	
RWTComMsg.rc	✓ up-to-date	
StringtableVBA.ini	✓ up-to-date	
Textitems.resx	✓ up-to-date	

□ Force update for unchanged source files

A summary result of the update data process is sent to the *Output* tab in the *Miscellaneous* bar. Changed items are indicated to the user in several ways, described in the following section.



```
Creating XML data file...
File: C:\Program Files\RC-WinTrans X8\Samples\Translation\Project1\Scribble.rc
Successful.

Translation data update for file "Scribble.rc" ( Japanese ) was successful.
0 new item(s).
2 changed item(s).
0 deleted item(s).
```

Figure 2: The *Output* tab view showing a summary of the results of an update process.

3.4.3 Reviewing Changes in the Updated Project

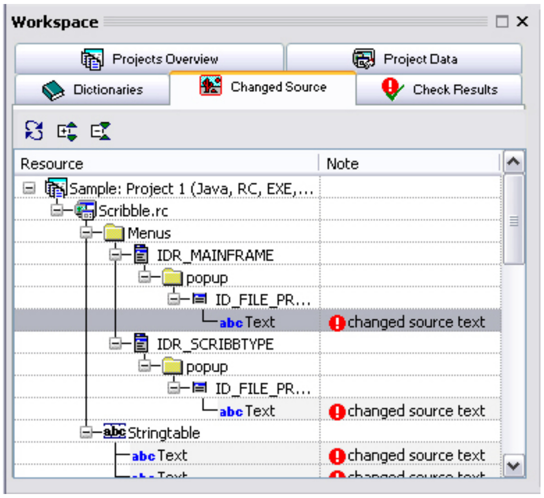
The database update process marks all changed items with the status “changed” (“changed” flag) in the translation database. The “changed” flag remains attached to a translation unit permantly until the the target data is confirmed. Once a translation is confirmed, RC-WinTrans removes the “changed” flag and changes the state to “translated.”

Visually, translation units for which the source text has changed are indicated to the user in the following ways:

- ***Changed Source* tab view** (Figure 1): Provides a list of all items which have changed since the translation project was updated. This view lets you see all the affected resources at a glance, and clicking on an item in the list selects it in both the *Text Table* view and the *Translation Edit* bar where it can be translated, reviewed and/or modified. (See also Section 3.3.3.4.)
- ***Text Table* view:**
  1. Red font is used for the target texts of changed items to make them easier to find in the *Text Table* view (Figure 2). After the target text has been confirmed the font color will revert back to the one selected on the *Text Table Colors* tab (*Options* dialog box).
  2. The colored status indication (column: “Status”) in the *Text Table* view (Figure 2) helps you to quickly identify any changed items needing to be reviewed and/or modified (by marking those items with the status color explicitly specified for changed items).
  3. An *Alert* symbol (❗) and the text “changed source” appear in the “Comment” column of affected items.
- ***Translation Edit* bar:** Displays changes made to a selected translation unit in the *(Previous)* and *(Current) Source* fields where the exact changes are highlighted in yellow (Figure 3). The standard changes indicator, the red dot marker, is also shown here.

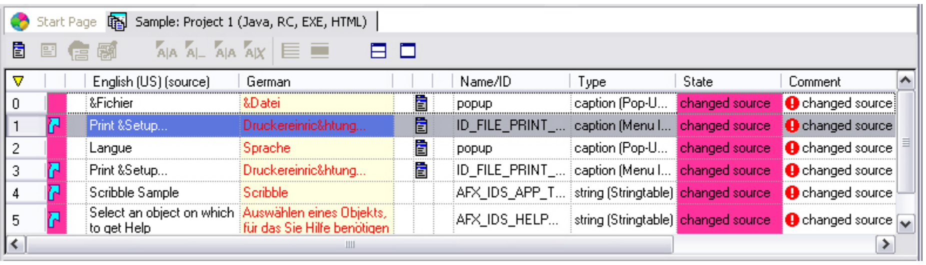
New (non-translated) items are marked in the *Text Table* view with a specific status indication color and the text “new” in the “State” column.

Figure 1: The *Workspace* bar's *Changed Source* view showing a list of changed items found during the update process.

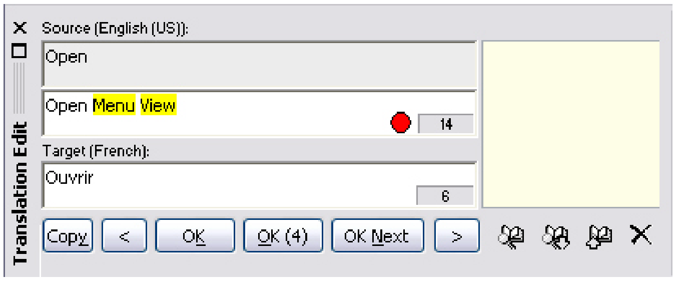


Resource	Note
Sample: Project 1 (Java, RC, EXE,…	
Scribble.rc	
Menus	
IDR_MAINFRAME	
popup	
ID_FILE_PR…	
abc Text	changed source text
IDR_SCRIBBTYPE	
popup	
ID_FILE_PR…	
abc Text	changed source text
Stringtable	
abc Text	changed source text

Figure 2: Changed items indication in the *Text Table* view.



		English (US) (source)	German		Name/ID	Type	State	Comment
0		&Fichier	&Datei		popup	caption (Pop-U…	changed source	changed source
1		Print &Setup...	Druckereinric&htung...		ID_FILE_PRINT_...	caption (Menu I...	changed source	changed source
2		Langue	Sprache		popup	caption (Pop-U...	changed source	changed source
3		Print &Setup...	Druckereinric&htung...		ID_FILE_PRINT_...	caption (Menu I...	changed source	changed source
4		Scribble Sample	Scribble		AFX_IDS_APP_T...	string (Stringtable)	changed source	changed source
5		Select an object on which to get Help	Auswählen eines Objekts, für das Sie Hilfe benötigen		AFX_IDS_HELP...	string (Stringtable)	changed source	changed source



Source (English (US)):

Open

Open Menu View 14

Target (French):

Ouvrir 6

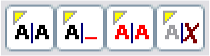
Figure 3: Changed items indication in the *Translation Edit* bar. The *(Previous) Source* field at the top shows the source text before it was modified (here: "Open"). The *(Current) Source* edit field just below it shows the source text as it is now (here: "Open Menu View"). Note that the differences between the two texts are highlighted in the *(Current) Source* field.

Filtering Changed Items

There are three different ways that changed items can be isolated (shown while excluding all other status types) in the *Text Table* view:

1. Changed items are automatically isolated in the *Text Table* view when you click on an item listed in the *Changed Source* view (*Workspace* bar).

2. Use the filtering toolbar buttons in the *Project* window (shown right) to show changed items only (*Show Items with Changed Source* button)/hide translated and non-translated items.



3. To select and isolate changed items, use the *Select by State* | *Changed Source* command from the *Edit* menu. Then press the *Show Selected Only* toolbar button (shown above, left) in the *Project* window.

3.5 Translation Tools and Features

3.5.1 Check Functions

RC-WinTrans comes equipped with a set of check functions to help ensure that your localized software displays correctly and runs properly. This section describes the individual check functions and, in some cases, provides tips for correcting found problems.

Running Checks

The *Checks* toolbar (Figure 1) provides quick access to the different check functions found in RC-WinTrans. Run a check (or a combination of different checks (“multi-check”)) by clicking on the appropriate toolbar button or by selecting the same command from the *Checks* menu (Figure 2).

Checks are performed for the entire project (all resources) if the *Project* window is the view that is currently active; if the *Dialog Box* or the *Menu* view is active, the check(s) will only be performed for the selected dialog box or menu, respectively. Generally, checks will be performed for all the data in the currently active project.

Some checks are performed automatically, such as the hotkey check (for menus and dialog boxes), the text fit check (dialog boxes), and the overlapping items check (dialog boxes).

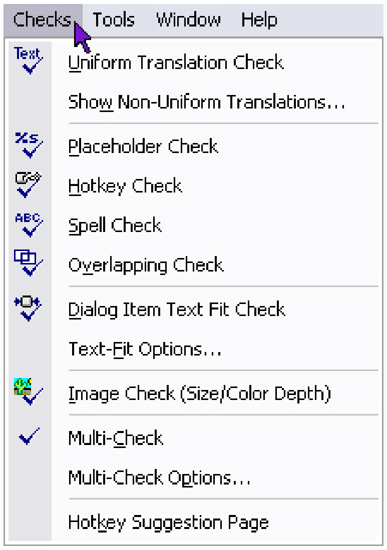
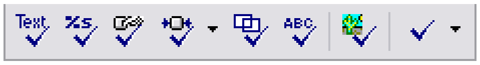


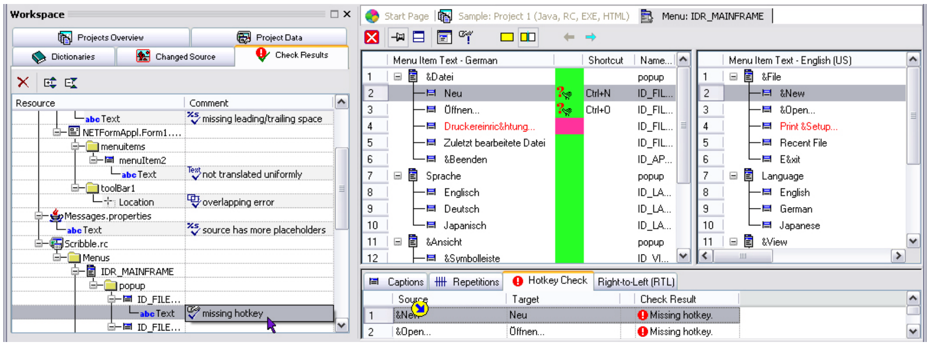
Figure 1: The *Checks* menu.

Figure 2: The *Checks* toolbar.



Display of Check Results

Anytime a check is run manually as described above, the results are sent to the *Check Results* tab view in the *Workspace* bar (Section 3.3.3.5). Not only does the *Check Results* tab provide you with a summary of all detected errors, but a simple mouse click on one of the listed items will automatically select that item in the most appropriate view (*Dialog Box*/*Menu*/*Text Table* view/*Translation Edit* bar) for reviewing and correcting the problem.



Resource	Comment
Text	missing leading/trailing space
NETFormAppl.Form1....	
menuitems	
menuItem2	
Text	not translated uniformly
toolBar1	
Location	overlapping error
Messages.properties	
Text	source has more placeholders
Scribble.rc	
Menus	
IDR_MAINFRAME	
popup	
ID_FILE...	
Text	missing hotkey
ID_FILE...	

	Menu Item Text - German		Shortcut	Name...
1	&Datei			popup
2	Neu		Ctrl+N	ID_FIL...
3	Öffnen...		Ctrl+O	ID_FIL...
4	Druckereinric&htung...			ID_FIL...
5	Zuletzt bearbeitete Datei			ID_FIL...
6	&Beenden			ID_AP...
7	Sprache			popup
8	Englisch			ID_LA...
9	Deutsch			ID_LA...
10	Japanisch			ID_LA...
11	&Ansicht			popup
12	&Symbolleiste			ID_VI...

	Menu Item Text - English (US)
1	&File
2	&New
3	&Open...
4	Print &Setup...
5	Recent File
6	E&xit
7	Language
8	English
9	German
10	Japanese
11	&View

	Source	Target	Check Result
1	&New	Neu	Missing hotkey.
2	&Open...	Öffnen...	Missing hotkey.

Figure 3: The *Check Results* tab view (left) showing the results of a multi-check operation. Clicking on an entry (a menu item has been selected here) in the list of results automatically selects it in the most appropriate view for handling that specificresource type/found error (in this case the *Menu* view, shown right). Note that the item has been marked and selected in the *Hotkey Check* tab (bottom right).

3.5.1.1 Spell Check

RC-WinTrans has a spell check capability for western languages with various options for checking (and correcting) the spelling of translation texts.

A spell check can be performed for a single word or single resource text entry and via the appropriate check function performed for all of the "string type" resource types in the current project at once. The spell check is also performed automatically whenever RC-WinTrans generates a report (HTML document) for a translation project (see Section 3.5.2.2, "Reports," for more information).

NOTE: The spell check function is **only available for Western languages**. It is **not** available for Eastern European, Asian, or Arabic languages.

RC-WinTrans has two spell check features: (1) an Internal Spell Checker and (2) access to the spell check function in MS Word (if installed).

Internal Spell Checker

The Internal Spell Checker checks for spelling errors in 12 different languages using the spelling dictionaries included with the RC-WinTrans installation.

MS Word Spell Checker

A second (external) spell check is available by accessing the function from Microsoft Word (if this is installed). Versions from Office 97 and later are supported. RC-WinTrans can make use of all spelling dictionaries currently available for MS Word. These MS Word spelling dictionaries can be found in the common "Microsoft Shared" folder:

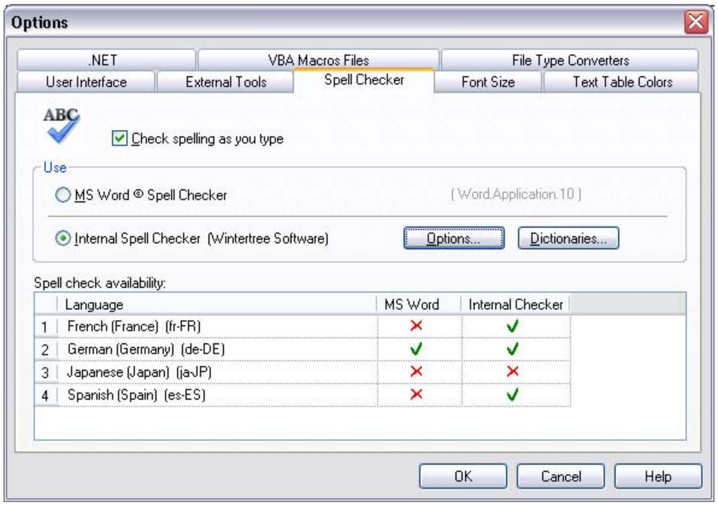
\Program File\Common Files\Microsoft Shared\Proof\

NOTE: MS Word is usually installed with only a few spelling dictionaries (approx. 4 languages). The languages of these dictionaries depends on the language of the MS Office pack that is installed. Additional dictionaries for other languages can be used by copying the files (.lex + .dll file) to the shared folder (given above).

You can see whether or not spell check is supported (dictionaries are available) for an RC-WinTrans project target language on the *Spell Checker* tab (*Options* dialog box).

Spell Check Settings and Options

To set the active Spell Checker (Internal or the one from MS Word) open the *Options* dialog box and select the *Spell Checker* tab found there (Figure 1). This page also indicates the availability of the spell check(s) for each of the target languages in the active translation project. Use the *Options* button to adjust the settings for the Internal Spell Checker via *the Spell Checker Options* dialog box (Figure 2).



☑ Check spelling as you type

Use

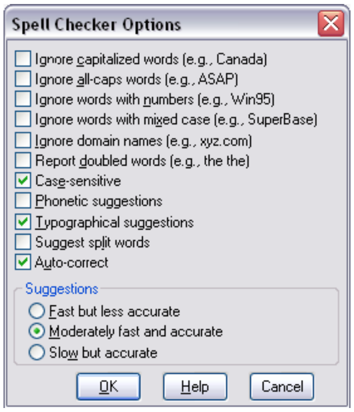
○ MS Word ® Spell Checker (Word.Application.10)

● Internal Spell Checker (Wintertree Software)

Spell check availability:

	Language	MS Word	Internal Checker
1	French (France) (fr-FR)	✗	✓
2	German (Germany) (de-DE)	✓	✓
3	Japanese (Japan) (ja-JP)	✗	✗
4	Spanish (Spain) (es-ES)	✗	✓

Figure 1: The *Spell Checker* tab active in the *Options* dialog box.



□ Ignore capitalized words (e.g., Canada)
□ Ignore all-caps words (e.g., ASAP)
□ Ignore words with numbers (e.g., Win95)
□ Ignore words with mixed case (e.g., SuperBase)
□ Ignore domain names (e.g., xyz.com)
□ Report doubled words (e.g., the the)
☑ Case-sensitive
□ Phonetic suggestions
☑ Typographical suggestions
□ Suggest split words
☑ Auto-correct

Suggestions

○ Fast but less accurate
● Moderately fast and accurate
○ Slow but accurate

Figure 2: The *Spell Checker Options* dialog box.

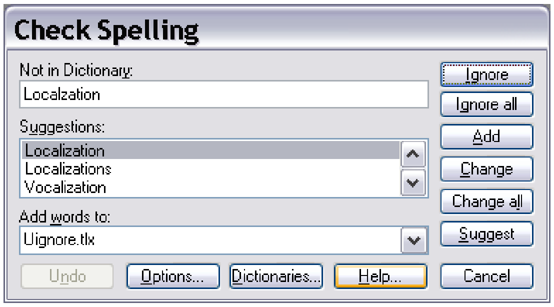
Internal Spell Checker Dictionaries

The dictionaries used by the Internal Spell Checker are managed on the *Internal Spell Checker Dictionaries* property page, accessed by clicking on the *Dictionaries* button on the *Spell Checker* tab (Figure 1). Here existing dictionaries can be assigned to a language, new user dictionaries can be created, and the contents of a user dictionary can be edited.

For more detailed information on Spell Checker dictionaries please refer to the RC-WinTrans Help (“Spell Check Dictionaries” and “Internal Spell Checker Dictionaries” topics).

Manual Spell Check (Individual Items)

Check the spelling of the currently selected translation unit or of a selected part of it by using the *Check Spelling* command button in the *Translation Edit* bar. If the Spell Checker finds a (possible) spelling error, the *Check Spelling* dialog box (Figure 3) is opened to correct the word. If the word is written correctly a (green) *Flash Info* window will appear informing you of this.



Check Spelling

Not in Dictionary: Localzation

Suggestions:
Localization
Localizations
Vocalization

Add words to: Uignore.tlx

Figure 3: Suggested corrections for a misspelled word in the *Check Spelling* dialog box.

Automatic Spell Check While Entering Text ("As You Type")

An automatic spell check can be enabled to alert the user to errors while the translation text is being entered/edited in the *Target* edit field of the *Translation Edit* bar. A (possible) misspelling is shown in red and underlined characters. Right mouse-click on the word to call up a local menu with a list of suggestions for possible corrections. Select the correct word from the suggestion list to replace the word in the *Target* field.

The automatic spell check is enabled by default for languages for which a spell check (and Spell Checker dictionaries) is (are) available. The automatic spell check can be disabled on the *Spell Checker* tab in the *Options* dialog box (Figure 1).

Adding New Entries to a Spelling Dictionary

To add a word to the user spelling dictionary, press the *Check Spelling* command button in the *Translation Edit* bar. The *Check Spelling* dialog box (Figure 3) opens where the *Dictionaries* button accesses the commands to do this.

Spell Check for All Text

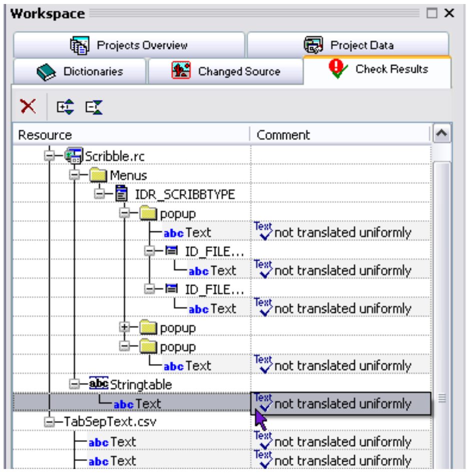
The spelling of all the strings in the current translation project can be checked by using the *Spell Check* toolbar button (shown left) or by selecting the *Spell Check* command from the *Checks* menu. The results are sent to the *Check Results* view where a single item in the list can be selected to display and/or edit it in the *Text Table* view or *Translation Edit* bar. The check can also be performed as part of a multi-check operation (selected checks run simultaneously- see Section 3.5.1.12 of this guide for more information).

3.5.1.2 Uniform Translation Check

This function checks all translation units (of the type “text”) having identical source texts and determines whether they have all been translated the same way. Found problems are items that have a different translation than items with the exact same source text. The ampersand character (&) is ignored when the source and translation texts are compared by this check.

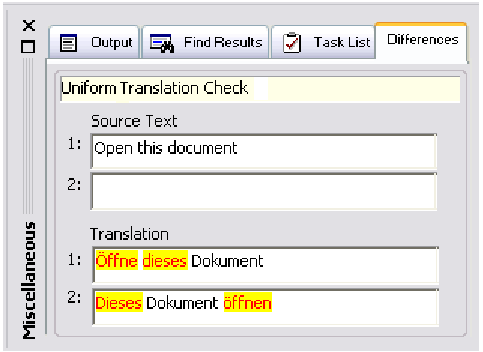
Comparing Different Translations

Compare the differing translation texts reported for an item in the *Different Translations* tab view where the texts are displayed and the differences highlighted. Select an item for which the error has been reported in the *Check Results* tab view (Figure 1) and the *Different Translations* tab will be activated in the *Miscellaneous* bar (Figure 2), where the differences can be reviewed.



Resource	Comment
Scribble.rc	
Menus	
IDR_SCRIBBTYPE	
popup	
abc Text	not translated uniformly
ID_FILE...	
abc Text	not translated uniformly
ID_FILE...	
abc Text	not translated uniformly
popup	
popup	
abc Text	not translated uniformly
abc Stringtable	
abc Text	not translated uniformly
TabSepText.csv	
abc Text	not translated uniformly
abc Text	not translated uniformly

Figure 1: The *Check Results* tab view showing several uniform translation errors. Select an item from the list to review inconsistencies in the *Differences* tab view (Figure 2).



Uniform Translation Check

Source Text

1: Open this document

2:

Translation

1: Öffne dieses Dokument

2: Dieses Dokument öffnen

Figure 2: The *Differences* tab view where differing translations (for the same source text) are displayed.

Different Translations Allowed

A translation unit can be designated as having a “unique translation,” (symbol: 1x) and thereby excluded from being considered and reported by this check. See the “‘Unique Translation’ Flag” section in the “Translating Identical Source Texts (Repetitions)” topic in the RC-WinTrans Help for more information.

3.5.1.3 Hotkey/Shortcut Key Check

This function performs the following two actions for a dialog box or menu:

1. Checks to see if the target language text contains a hotkey for each occurrence of a hotkey in the source language text. The source and target language hotkeys may differ.

   The term “hotkey” is used to indicate a notation marking a particular character for short access to a dialog box or menu item by striking this character key. *Example:* Strike the “**F**” key to access the “**File**” menu item.

2. Checks each “local group” (a full dialog box or single pop-up menu) to see if the hotkeys used in the target language are unique within this group. More than one occurrence of the same hotkey will be reported.

See also the RC-WinTrans Help topic “Hotkeys – Suggestions and Conflict Resolution.”

Doubled and missing hotkeys for **menu resources** are automatically listed in the *Menu* view’s *Hotkey Check* tab- a check does not need to be run manually. Doubled and missing hotkeys are immediately indicated in the menu tree view by the following symbols:

Missing Hotkey

This symbol indicates that the source text for the item contains a hotkey but the translation text does not.

The symbol is used in in the *Menu* view (Figure 1.)

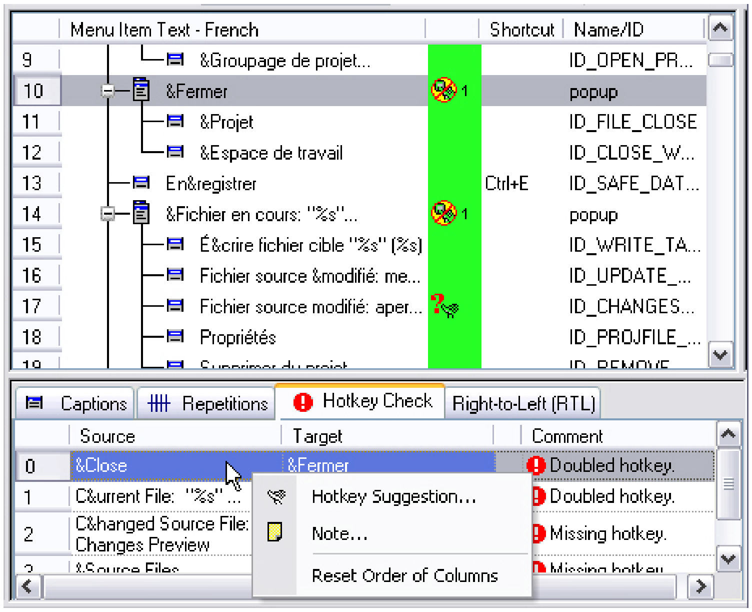
1
1

Doubled Hotkeys

The appearance of this symbol indicates that the same hotkey has been assigned to two or more menu items. The number next to the symbol shows which items share the same hotkey.

The symbols are used in in the *Menu* view (Figure 1.)

As with menu resources, doubled and missing hotkeys for **dialog box resources** are automatically listed in the *Dialog Box* view's *Hotkey Check* tab so that there is no need to run the check manually.



	Menu Item Text - French		Shortcut	Name/ID
9	&Groupage de projet...			ID_OPEN_PR...
10	&Fermer	1		popup
11	&Projet			ID_FILE_CLOSE
12	&Espace de travail			ID_CLOSE_W...
13	En&registrer		Ctrl+E	ID_SAFE_DAT...
14	&Fichier en cours: "%s"...	1		popup
15	É&crire fichier cible "%s" (%s)			ID_WRITE_TA...
16	Fichier source &modifié: me...			ID_UPDATE_...
17	Fichier source modifié: aper...	?		ID_CHANGES...
18	Propriétés			ID_PROJFILE_...
19	Supprimer du projet			ID_REMOVE

	Source	Target	Comment
0	&Close	&Fermer	Doubled hotkey.
1	C&urrent File: "%s" ...		Doubled hotkey.
2	C&hanged Source File: Changes Preview		Missing hotkey.
3	&Source Files		Missing hotkey.

Figure 1: The *Menu* view with hotkey errors indicated in the menu tree view at the top and the *Hotkey Check* tab view active below it. Here the menu item in line 17 is missing a hotkey while a doubled hotkey ("&F") exists for the two menu items in lines 10 and 14. The *Hotkey Suggestion* command can be seen in the local menu that has been opened (bottom center).

Resolving Hotkey Errors

Hotkey errors can be resolved in one of the following manners:

1. Edit the target text in the *Translation Edit* bar or use direct inline editing in either the *Text Table* or *Repetitions* view to place an ampersand character (& – hotkey indicator) in front of the letter you want to act as a hotkey.

2. Use the *Hotkey Suggestion* dialog box (described further on) to select a hotkey from an automatically-generated list of suitable choices. Open this dialog with a right mouse-click (anywhere in the view) to open a local menu where the *Hotkey Suggestion* command is located. This command is also accessible via RC-WinTrans' *Edit* menu. The *Hotkey Suggestion* button in the *Menu* view serves the same purpose.

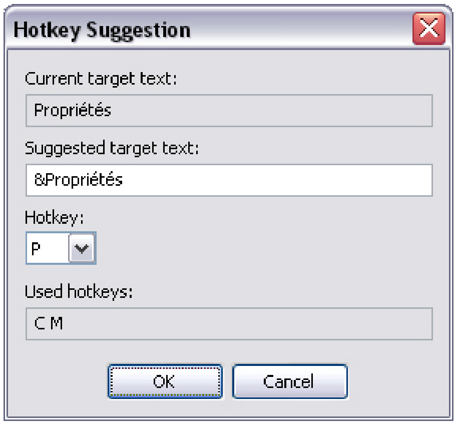
***Hotkey Suggestion* Dialog Box**

The *Hotkey Suggestion* dialog box (Figure 2) offers suggestions for a hotkey that would be suited to the currently selected menu (or dialog box) text. There are three different ways to **access the dialog- from the *Menu* view:**

1. Right mouse-click anywhere in the view to open a local menu and select the *Hotkey Suggestion* command.
2. Click on the *Hotkey Suggestion* toolbar button located at the top of the *Menu* view.
3. Use the *Hotkey Suggestion* command from the *Edit* menu.

To access the *Hotkey Suggestion* dialog box **from the *Dialog Box* view**, use one of the following methods:

1. Right mouse-click anywhere in the view to open a local menu and select the *Hotkey Suggestion* command, **OR**
2. Use the *Hotkey Suggestion* command from the *Edit* menu.



Current target text: Propriétés

Suggested target text: &Propriétés

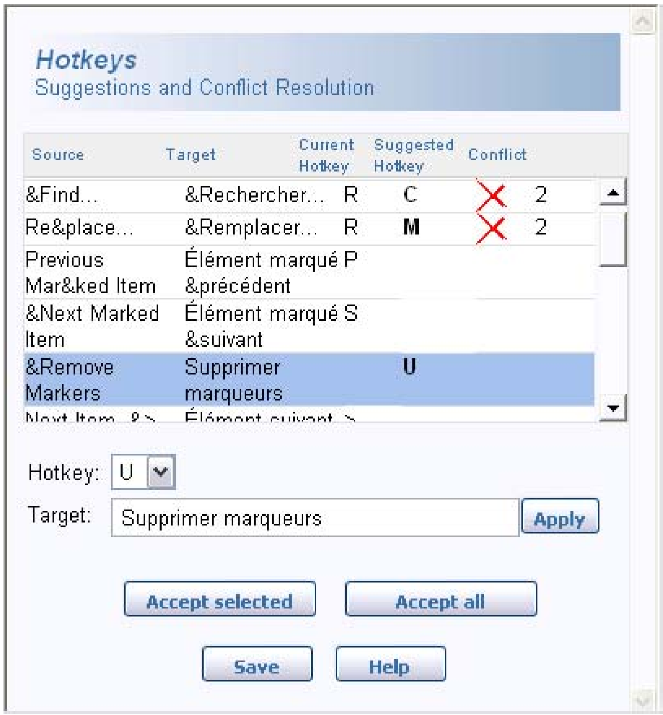
Hotkey: P

Used hotkeys: C M

Figure 2: The *Hotkey Suggestion* dialog box. Here the hotkey for the French target text “Propriétés” is missing and the letter “P” is being suggested for use.

Hotkeys: Suggestions and Conflict Resolution Page

An additional resource for handling hotkeys is the *Hotkeys: Suggestions and Conflict Resolution* HTML page (Figure 3). This page (opened from the *Checks* menu) shows you all the hotkey characters currently assigned within the selected menu or dialog box as well as any hotkey errors which have been found along with options for resolving these.



Hotkeys
Suggestions and Conflict Resolution

Source	Target	Current Hotkey	Suggested Hotkey	Conflict
&Find...	&Rechercher...	R	C	✗ 2
Re&place...	&Remplacer...	R	**M**	✗ 2
Previous Mar&ked Item	Élément marqué &précédent	P		
&Next Marked Item	Élément marqué &suivant	S		
&Remove Markers	Supprimer marqueurs		**U**	
Next Item &>	Élément suivant >			

Hotkey: U

Target: Supprimer marqueurs

Figure 3: The *Hotkeys: Suggestion and Conflict Resolution* page showing conflicts between assigned hotkeys and a missing hotkey for the selected item. The letter “U” is being suggested for use.

Hotkey Check for Detached Strings

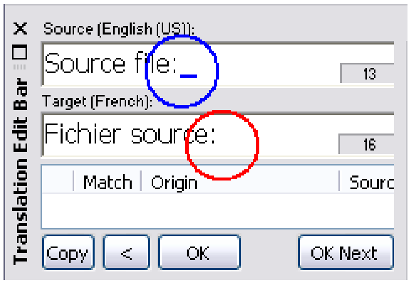
This function checks each detached string for a missing hotkey (“detached”: a single string, not a string in a dialog box or menu resource; usually a string in a Win32 RC/EXE or .NET **stringtable**). An error will be reported if the source language text has a hotkey (i.e., a single ampersand character (&) in front of a non-whitespace character) but the target text does not.

Run the text fit check for detached strings via the multi-check (described in Section 3.5.1.12).

3.5.1.4 Leading/Trailing Spaces Check

A common problem in software localization is ensuring the uniformity of source and target text with respect to leading and trailing spaces. If a given source text has a certain number of leading/trailing spaces, this function verifies whether the target text has the same number of leading/trailing spaces as is present in the source text.

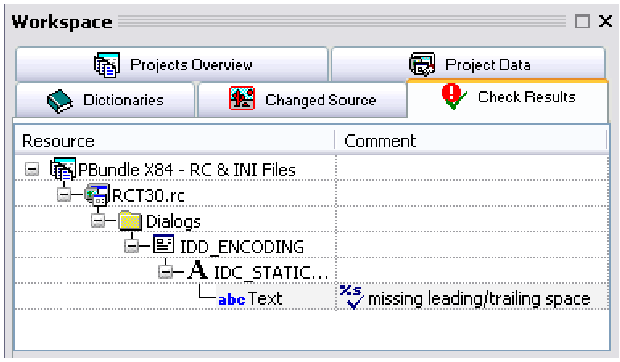
NOTE: This function also applies to **whitespace characters** (particularly tab characters).



Source (English (US)):
Source file:_ 13

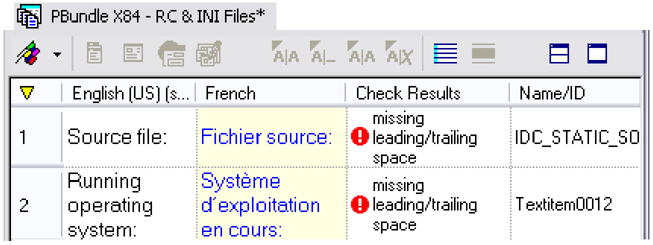
Target (French):
Fichier source: 16

Figure 1: The *Translation Edit* bar showing a translation unit with a trailing space error. The target text is missing the trailing space that is present in the source text (indicated in the display by the underline mark.



Resource	Comment
PBundle X84 - RC & INI Files	
RCT30.rc	
Dialogs	
IDD_ENCODING	
IDC_STATIC...	
abc Text	missing leading/trailing space

Figure 2: A leading/trailing space error reported for a dialog box item text in *the Check Results* tab view.



▽	English (US) (s...	French	Check Results	Name/ID
1	Source file:	Fichier source:	missing leading/trailing space	IDC_STATIC_SO
2	Running operating system:	Système d´exploitation en cours:	missing leading/trailing space	Textitem0012

Figure 3: Indication of leading/trailing space errors in the *Project* window's *Text Table* view.

3.5.1.5 Trailing Colon/Ellipsis Check

This function checks for each occurrence of a colon (:) or ellipsis (three dots (...)) at the end of a source language string. In such cases the check determines whether the target language text also ends with a colon or an ellipsis.

3.5.1.6 Placeholder Check

For each occurrence of a placeholder (e.g., {1}, {2}, %2, %s) in a source text, RC-WinTrans checks to see if the target text has the same placeholder). **This check can only be performed for Win32 files, .NET files, and Java properties files-** all other file types are excluded.

Two types of placeholders are checked in Win32 resource files:

1. **printf**-type placeholders (e.g., "%**s**," "%**lx**," "%**.1d**"...)
2. **FormatMessage**-type placeholders (e.g., "%**1**," "%**2**")

While the **printf**-type placeholders need to be **equal** with respect to their **position** within the format string as well as to their format specification (a "%s" in the source language text must have a corresponding "%s" in the target language text), the **FormatMessage**-type placeholders are more flexible. These types of placeholders come with a position specification, so simply checking for the presence of each placeholder in both the source and the target language texts is sufficient.

3.5.1.7 Control Characters Check

This function searches for any occurrence of the C-language-type control sequences "**\t**" and "**\n**" (indicating a horizontal tab and a line break, respectively). For all such occurrences, the control character check makes sure that the same control character with the same ordering is present in the target language.

3.5.1.8 Text Fit Check (Dialog Box Controls)

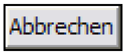
Text Fit Check for Win32 Dialog Box Controls and .NET Forms

The text fit check is used to determine whether a dialog box control is large enough (width/height) to display the entire target language text.

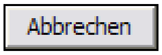
NOTE: You can exclude a dialog box control from the text fit check on the *Text Fit* tab in the *Dialog Box* view (*Enable check* checkbox). It can also be excluded on the *Checks* tab of the *Dialog Box Control Properties* dialog box (click on the *Properties* button or right mouse-click in the *Dialog Box* view to open).

The text fit check is equipped with settings for **extra space** required in the horizontal (x) and/or vertical (y) directions. For positive values, the control's size has to be the given number of pixels larger in order to fit. Negative values will result in a less critical test in that each control may be the given number of pixels smaller, i.e., a negative value accepts an overhang of the text.

Example:

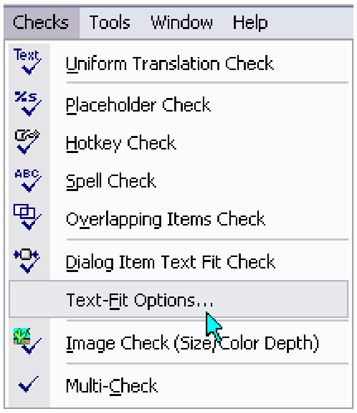


Minimum item width with the additional space value set to "**0**." The text fit check will verify that the text fits within the left and right borders of the control.

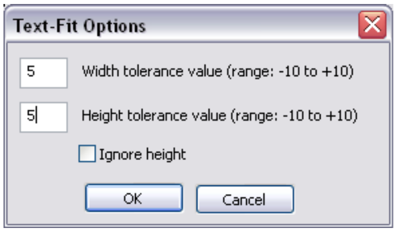


Minimum item width with the additional space value set to "**10**." The text fit check will verify that the text fits within the left and right borders of the control with a distance of 5 (unit: DLU (Win32), pixel (.NET)) on both sides.

Both values are set in the *Text Fit Options* dialog box. Open this dialog by clicking on the *Settings* button on the *Text Fit* tab located at the bottom of the *Dialog Box* view. Only values in the range -10 – +10 are allowed.



Figures 1 (left) and 2 (below): Use the *Text Fit Options* command in the *Checks* menu to open the *Text Fit Options* dialog box where you can specify the amount of (extra) space that a dialog box item should have between its text and the borders of the control in order to be approved by the text fit check.



Text-Fit Options

5 Width tolerance value (range: -10 to +10)

5 Height tolerance value (range: -10 to +10)

□ Ignore height

3.5.1.9 String Length Check

Maximum String Length Check

This function checks for each occurrence of a colon (:) or ellipsis (three dots, (...)) at the end of a source language string. In such cases the check determines whether the target language text also ends with a colon or an ellipsis.

This function checks to see if a string exceeds a specified maximum string length. The *Maximum String Length* dialog box (Figure 3) is used to specify the maximum string length and to enable the check for a string item.

There are **three different units** that can be used when defining the maximum string length:

1. **Number of characters:** The total number of characters allowed for the target text.
2. **Expansion percentage:** Given amount (expressed in a percentage) of extra length allowed for the target text.
3. **Dialog units (Win32):** The maximum length of the target text expressed in (Win32) dialog units (DLUs).

Run the check using the *Maximum String Length Check* button in the *Checks* toolbar (Figure 1) or by selecting the same command from the *Checks* menu. As for all check functions, the results of a maximum string length check are displayed in the *Check Results* tab view (*Workspace* bar).

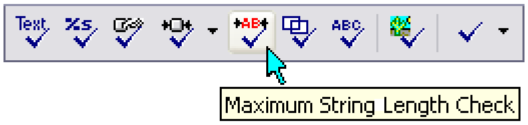


Figure 1: The *Maximum String Length Check* button in the *Checks* toolbar.

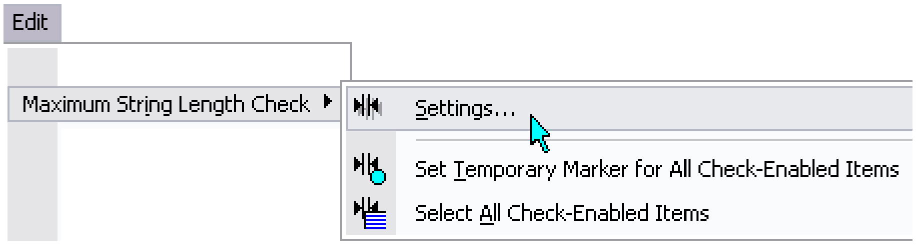
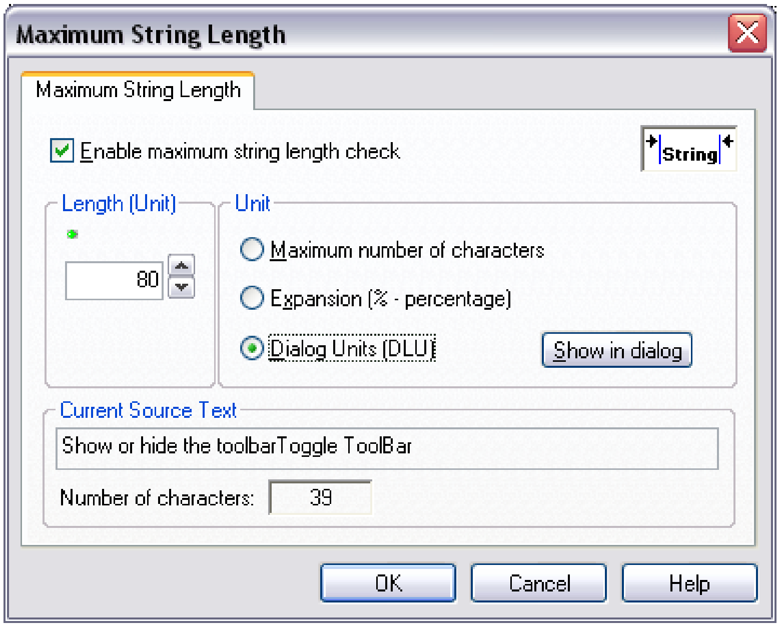

Edit
Maximum String Length Check
Settings...
Set Temporary Marker for All Check-Enabled Items
Select All Check-Enabled Items

Figure 2: The *Edit* menu and the command (*Settings*) for opening the *Maximum String Length* dialog box (for one or a number of selected items).

You can determine/locate all strings that have a maximum string length defined and that are enabled for the check with the following two commands from the *Edit* menu (see Figure 2):

- ***Set Temporary Marker for All Check-Enabled Items***
  Sets a temporary (blue dot) marker (●) next to all items that have been enabled for the maximum string length check.
- ***Select All Check-Enabled Items***
  Selects all items (in the *Text Table* view) that have been enabled for the maximum string length check.



Maximum String Length

Maximum String Length

☑ Enable maximum string length check

Length (Unit): 80

Unit:
○ Maximum number of characters
○ Expansion (% - percentage)
● Dialog Units (DLU)

Current Source Text: Show or hide the toolbarToggle ToolBar

Number of characters: 39

Figure 3: The *Maximum String Length* dialog box used to specify the maximum string length and enable the maximum string length check function for one or a number of selected target texts. Open the dialog box by selecting the *Maximum String Length Check* | *Settings* command from the *Edit* menu (Figure 2).

3.5.1.10 Overlapping Items Check

As with the text fit check, this function applies to **Win32 dialog box** controls and **.NET Forms** controls only. The overlapping check determines whether any of the controls overlap each other.

As with the text fit check, this function applies to **Win32 dialog box** controls and **.NET Forms** controls only. The overlapping check determines whether any of the controls overlap each other.

NOTE: A control that lies completely inside another control will be assumed valid and will not result in an overlapping error. Dialog box controls can be excluded from the overlapping items check, but it is important to remember that in some cases controls may overlap by design (dialog box layout) for a useful or technical reason. In such cases the control should be excluded from the check so that an overlapping error won't be reported. You can exclude a dialog box control from the overlapping items check on the *Overlapping* tab in the *Dialog Box* view (*Enable check* checkbox). It can also be excluded on the *Checks* tab of the *Dialog Box Control Properties* dialog box (click on the *Properties* button or right mouse-click in the *Dialog Box* view to open).

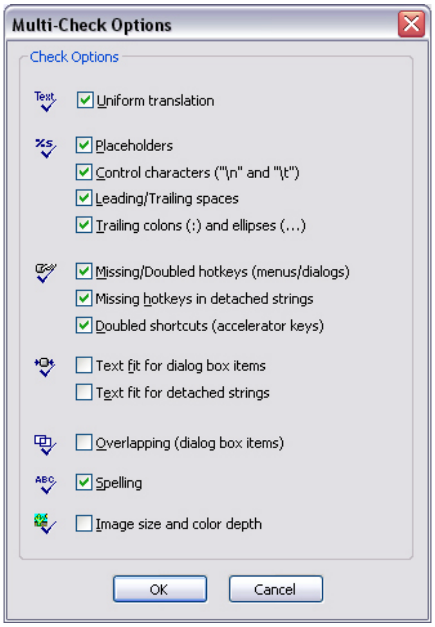
3.5.1.11 Image Check

Checks to see if the size and/or color depth of target images differ from the source images.

3.5.1.12 Multi-Check

RC-WinTrans' multi-check function performs a number of checks simultaneously. Select the checks to be included in the multi-check in the *Multi-Check Options* dialog box (figure follows). It can be opened by clicking on the *Multi-Check* toolbar button's drop-down button, or by selecting the *Multi-Check Options* command from the *Checks* menu.

Once the *Multi-Check Options* dialog box is open, click the checkboxes to select/unselect (include/exclude, respectively) individual checks. Confirm the selections by pressing the *OK* button, then run the multi-check function by clicking on (the main part of) the *Multi-Check* toolbar button.



Multi-Check Options

Check Options

☑ Uniform translation

☑ Placeholders
☑ Control characters ("\n" and "\t")
☑ Leading/Trailing spaces
☑ Trailing colons (:) and ellipses (...)

☑ Missing/Doubled hotkeys (menus/dialogs)
☑ Missing hotkeys in detached strings
☑ Doubled shortcuts (accelerator keys)

☐ Text fit for dialog box items
☐ Text fit for detached strings

☐ Overlapping (dialog box items)

☑ Spelling

☐ Image size and color depth

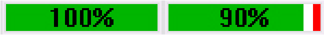
OK | Cancel

Figure: Various checks selected to be included in the next multi-check operation.

3.5.2 Reports and Statistics

3.5.2.1 Translation Statistics

***Status* Bar**



RC-WinTrans has two permanent translation status indicators in the *Status* bar located at the bottom of the main application window. The indicators show the translation status in percentages and as colored progress bars. The indicator on the left shows the amount translated in the active target language while the one on the right shows the amount of translated text displayed in the active view.

***Translation Status* Page**

The *Translation Status* page provides statistical information on the translation progress of a project or selected group. Open the *Translation Status* page with the *Statistics* command from the *Project* menu, or use the *Statistics* toolbar button in the *Project* toolbar (Figure 1).

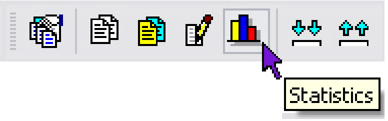
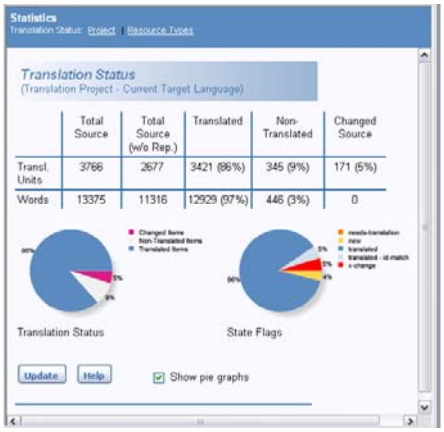


Figure 1: The *Statistics* button in the *Project* toolbar.



Statistics
Translation Status: Project | Resource Types

Translation Status
(Translation Project - Current Target Language)

	Total Source	Total Source (w/o Rep.)	Translated	Non-Translated	Changed Source
Transl. Units	3766	2677	3421 (86%)	345 (9%)	171 (5%)
Words	13375	11316	12929 (97%)	446 (3%)	0

Figure 2: Project statistics on the *Translation Status* page.

The translation status information is organized into a table. Statistics are given for all the translation units as well as for all the words contained in the current project or group.

Statistics Table

“Total Source” The total number of translation units/words contained in the current project or group.

“Total Source without Repetitions” The total number of translation units/words contained in the current project or group not including repeated entries (i.e., only one occurrence of identical entries is taken into account).

“Translated” The total number of translation units/words in the current project or group that have already been translated.

NOTE: “translated” means that an entry exists in the target text.

“Non-Translated” The total number of translation units/words in the current project or group for which no translation exists.

“Changed Source” The total number of translation units/words contained in the current project or group for which the source item has changed.

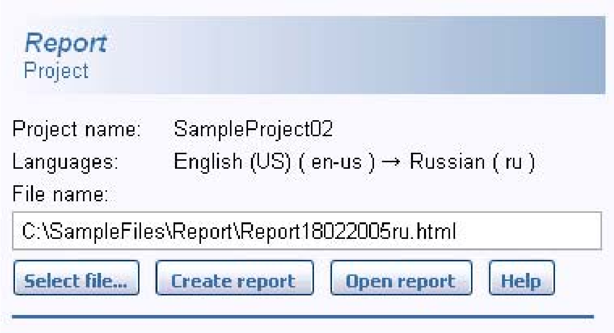
The translation status information table is supplemented by two pie charts which presents the project’s translation status to the user visually. The pie chart on the left shows the percentage of translated, non-translated, and changed items in the current project or group. The pie chart on the right shows information about the various states of the items contained in the current project or group.

Use the *Update* button near the bottom to refresh the information on the page. Checking the *Auto-update* checkbox tells RC-WinTrans to update the information on the page automatically.

3.5.2.2 Reports

Reports can be generated to provide an overview of the most important details of a translation project. The format of the report is an HTML page, making it ideal for saving or printing.

To create a report for the current project (and the current source and target language), click on the *Project Report* toolbar button (shown left), or use the *Report* command from the *Project* menu. The *Report* page that is activated contains the commands and file options for creating a project report.



Report
Project

Project name: SampleProject02
Languages: English (US) (en-us) → Russian (ru)
File name:
C:\SampleFiles\Report\Report18022005ru.html

Figure: The *Report* page for generating a project report.

The RC-WinTrans localization report will be called up automatically and displayed in a separate browser window. At the top of the report you will see the name of the project and its source and target languages, plus the date and time that the report was generated. This information is followed by a list of the source and target files contained in the project.

The next section of the report features a table with information on the project’s translation status. The table clearly displays the most important details about the words and translation units contained in the project, such as the total number of these, how many of them have already been translated, and how many have changed source text.

A project report will also list all of the re-check markers and tasks contained in the project. The names and locations of these items are provided here so that they can be easily found in the project.

3.5.3 Character Map

RC-WinTrans has a character map you can use to insert foreign characters into the target text while translating text in the *Translation Edit* bar. Open the character map by clicking on the *Character Map* button at the bottom of the *Translation Edit* bar (Figure 1).

Select a character by clicking on it with the mouse. The dialog box (Figure 2) will close and the selected character will be inserted into the *Target* edit field in the *Translation Edit* bar.

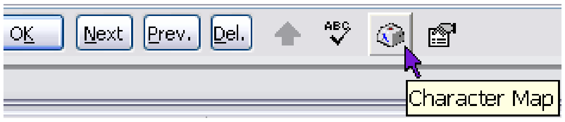


Figure 1: The *Character Map* button in the *Translation Edit* bar.

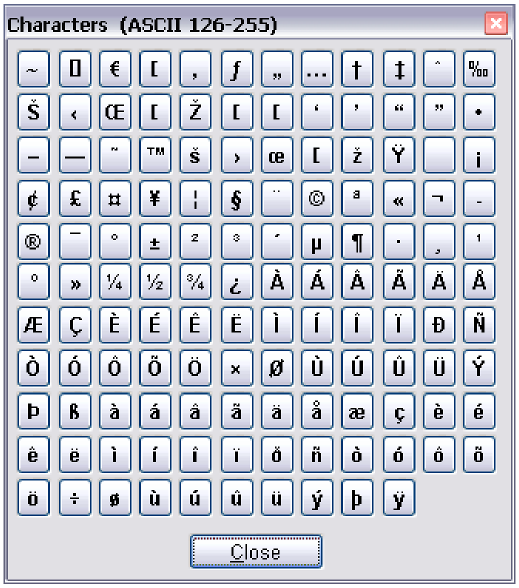


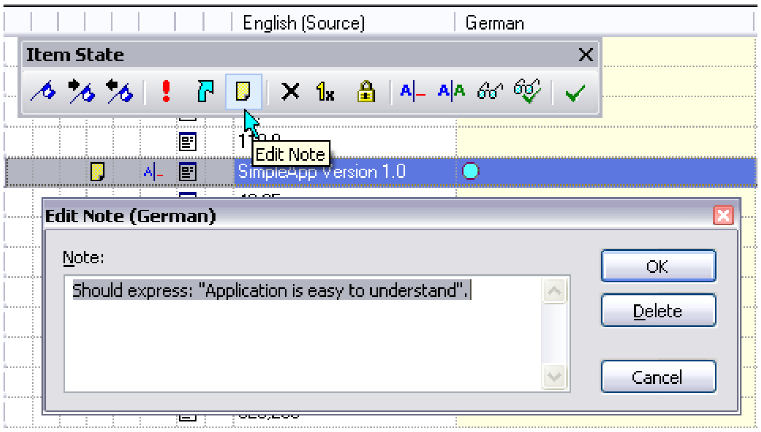
Figure 2: The *Characters* dialog box.

3.5.4 Notes

***Translation Edit* Bar / *Text Table* View**

Notes can be attached to the current translation unit by entering a comment in the *Note* edit field positioned at the top right corner of the *Translation Edit* bar. You can also use the *Edit Note* button in the *Item State* toolbar (Figure 1).

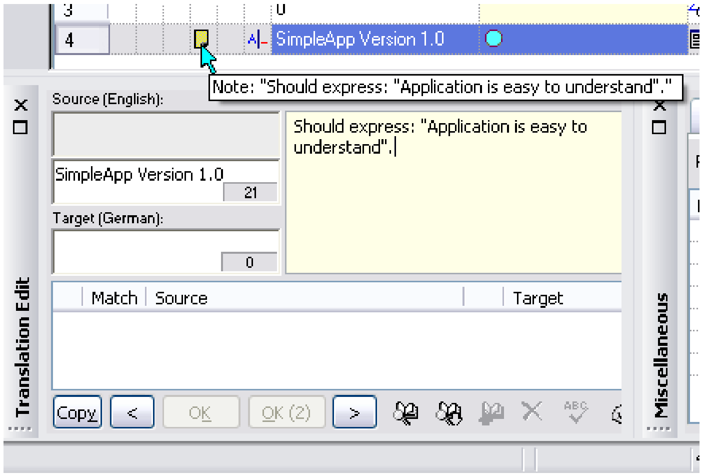
Figure 1: The *Edit Note* toolbar button and dialog box.



Note:

Should express: "Application is easy to understand".

The *Note* symbol (🗒) is displayed in the *Text Table* view next to items that have notes. The note is also displayed in a tooltip window if the mouse pointer is hovering over the *Note* symbol (Figure 2).



Note: "Should express: "Application is easy to understand"."

Source (English):

SimpleApp Version 1.0

21

Target (German):

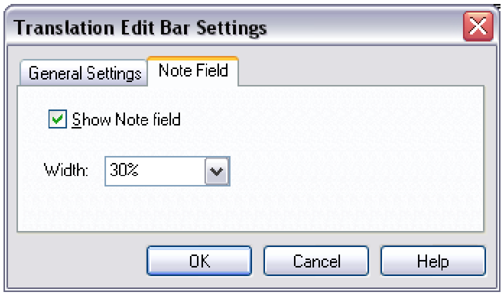
0

Should express: "Application is easy to understand".

Figure 2: The *Note* symbol and note text displayed as a tooltip in the *Text Table* view and as modifiable text in the *Translation Edit* bar.

NOTE: The size of the *Note* field can be adjusted- or, if desired, hidden from view- on the *Note Field* tab in the *Translation Edit Bar Settings* dialog box. This

dialog is opened with the Translation Edit Bar Settings button (shown left) at the bottom of the *Translation Edit* bar.



☑ Show Note field

Width: 30%

Figure 3: The *Note Field* tab in the *Translation Edit Bar Settings* dialog box.

***Project Data* View (*Workspace* Bar)**

Notes can also be assigned to whole files and/or groups (e.g., dialog boxes, menus, etc.) in the *Workspace* bar's *Project Data* view. Use the *Add Info* toolbar button or right mouse-click on the item and select the *Note* command from a local menu to enter/edit the note text.

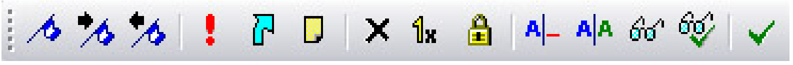
More information on the *Project Data* tab view can be found in Section 3.3.3.2 of this guide.

3.5.5 Bookmarks, Tasks, and Re-Check Markers

Bookmarks

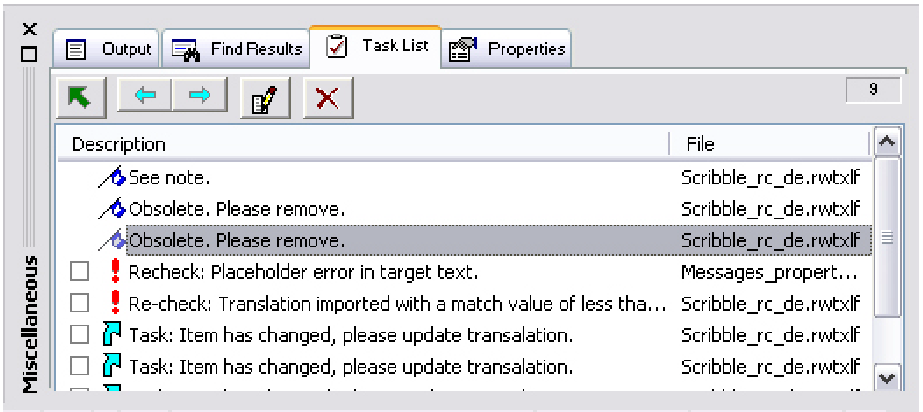
Bookmarks can be used to quickly and easily access resources that need to be referenced regularly. You can quickly assign a bookmark to or remove it from an item by clicking on the *Bookmark* button in the *Item State* toolbar (Figure 1). Move back and forth between bookmarks in the project with the two buttons next to it. The same commands are also available in the *Edit* menu. The *Edit* menu also has a *Clear All Bookmarks* command to remove all of the bookmarks in the project.

Figure 1: The *Item State* toolbar.



If a translation unit has a bookmark this is indicated in the *Text Table* view by the *Bookmark* symbol ().

All bookmarks used in a project are listed in the *Task List* tab view (*Miscellaneous* bar) where you can jump to a translation unit by double-clicking on its corresponding entry in the list. Each bookmark has a description text that can be edited in the *Task List* tab.



	Description	File
	See note.	Scribble_rc_de.rwtxlf
	Obsolete. Please remove.	Scribble_rc_de.rwtxlf
	Obsolete. Please remove.	Scribble_rc_de.rwtxlf
□	! Recheck: Placeholder error in target text.	Messages_propert...
□	! Re-check: Translation imported with a match value of less tha...	Scribble_rc_de.rwtxlf
□	Task: Item has changed, please update transalation.	Scribble_rc_de.rwtxlf
□	Task: Item has changed, please update transalation.	Scribble_rc_de.rwtxlf

Figure 2: Bookmarks, re-check markers, and tasks in the *Miscellaneous* bar's *Task List* tab view.

Tasks

Task markers (⚑) mark items with tasks in the *Text Table* view. A task is a specific job that needs to be done for the item bearing this symbol. Tasks and their description texts can be viewed in the *Miscellaneous* bar's *Task List* tab view.

Add or remove a task marker to/from an item by clicking on the *Task Marker* button in the *Item State* toolbar, or by selecting the *Task Marker* command from the *Edit* menu.

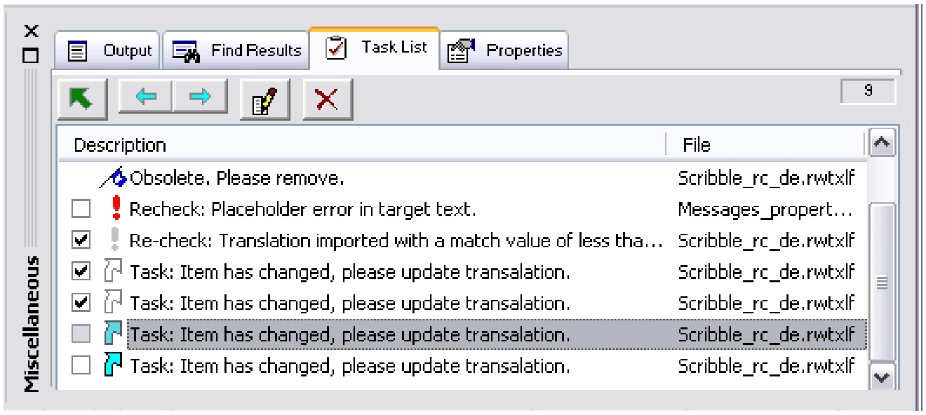
Once an item has been reviewed and nothing else needs to be done with it, you can indicate this by checking the checkbox next to the corresponding task marker in the *Task List* view (Figure 3). This will cause the task marker to turn from blue to white, indicating that the item has been reviewed and approved for the final version (translation).

Re-Check Markers

A re-check marker (!) is used to mark items needing to be reviewed. Re-check markers and their description texts are listed in the *Task List* tab view. Jump to an item in the *Text Table* view by double-clicking on its corresponding entry in the list.

Add or remove a re-check marker to/from an item by clicking on the *Re-Check Marker* button in the *Item State* toolbar, or use the *Re-Check Marker* command from the *Edit* menu.

Once an item has been reviewed and nothing else needs to be done with it, you can indicate this by checking the checkbox next to the corresponding re-check marker in the *Task List* tab (Figure 3). This will cause the re-check marker to turn from red to gray, indicating that the item has been reviewed and approved for the final version (translation).



	Description	File
	Obsolete. Please remove.	Scribble_rc_de.rwtxlf
☐	! Recheck: Placeholder error in target text.	Messages_propert...
☑	! Re-check: Translation imported with a match value of less tha...	Scribble_rc_de.rwtxlf
☑	Task: Item has changed, please update tansalation.	Scribble_rc_de.rwtxlf
☑	Task: Item has changed, please update tansalation.	Scribble_rc_de.rwtxlf
☐	Task: Item has changed, please update tansalation.	Scribble_rc_de.rwtxlf
☐	Task: Item has changed, please update tansalation.	Scribble_rc_de.rwtxlf

Figure 3: The *Task List* tab view showing some completed/reviewed tasks and re-check items, as indicated by the checked checkboxes and changed symbol color.

3.5.6 Temporary Markers

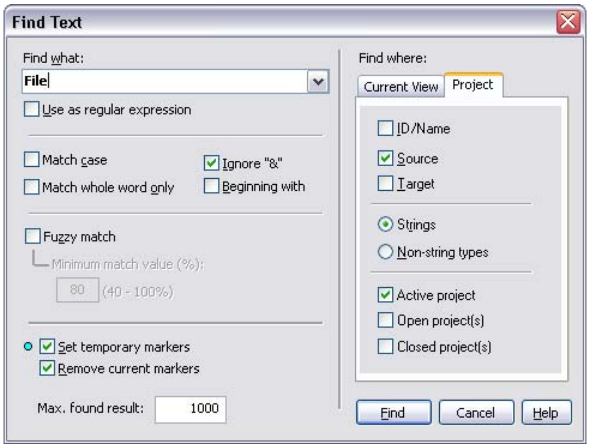
Temporary, or blue dot markers (●) are used to identify items that have been affected by an executed command or operation. Markers can also be set manually for a predefined set of filter options. The next operation that is performed will remove and replace the markers set by the previous one (exception: you can choose to leave the existing markers in place when performing a find text operation).

Move between marked items using the *Next Marked Item* and *Previous Marked Item* commands available in the main toolbar and in the *Edit* menu. Markers are displayed in the *Text Table*, *Menu*, and *Dialog Box* views

Figure: (left to right) The *Next Marked Item*, *Previous Marked Item*, and *Remove Markers* buttons in the *Main* toolbar.

3.5.7 Find Text Operation

Search for text in the currently active view by calling up the *Find Text* dialog (Figure 1) using the F3 function key, the *Find* toolbar button, or the *Find* command from the *Edit* menu.



Find what: File

□ Use as regular expression

□ Match case

☑ Ignore "&"

□ Match whole word only

□ Beginning with

□ Fuzzy match

Minimum match value (%): 80 (40 - 100%)

☑ Set temporary markers

☑ Remove current markers

Max. found result: 1000

Find where:

□ ID/Name

☑ Source

□ Target

● Strings

○ Non-string types

☑ Active project

□ Open project(s)

□ Closed project(s)

Figure 1: The *Find Text* dialog box.

***Find Text* Dialog Box**

- ***Find what* Field**
  Specifiy the text you’re looking for in the *Find what* edit field. You can also use the drop-down list to select from a list of up to 10 previous search strings.

- ***Use as regular expression* Checkbox**
  Check this option to use the text in the *Find what* field as a regular expression for the find text operation. Using regular expressions allows for more flexible find operations, as in the following two examples:

REGULAR EXPRESSION	DESCRIPTION
U(SA|K)	Match both strings: “USA” and “UK”
colou?r	Match both terms: “color” and “colour”

- ***Match case* Checkbox**
  Check this option to find only those text strings that match the case of the characters in the *Find what* field exactly. If this option is not selected, the command finds strings that match the characters in the *Find what* string regardless of whether the characters are upper- or lowercase.

- ***Match whole word only* Checkbox**
  Check this option to only match text strings that are preceded and followed by a space, tab, punctuation mark, or the beginning or end of a line. If this option is not selected, the command matches any string regardless of whether or not it is a fragment of a larger string.

- ***Ignore “&”* Checkbox**
  Checking this option tells RC-WinTrans to ignore ampersand mnemonic prefix characters (&) in texts.

- ***Beginning with* Checkbox**
  Check this option to force the beginning characters of the found results to match the text entered in the *Find what* field exactly.

- ***Fuzzy match* Checkbox and *Minimum match value* Edit Field**
  Check this option to include partially-matching strings in the found results. Enter a minimum match value (expressed in a percentage) in the field provided- strings must match the search string by at least the given percentage to be included in the found results.

  NOTE: Using this option to perform a search for a large number of text items or files can be quite time-consuming. It is intended to be used as a second search option when the main search (using the *Find what* field in combination with the *Match case*, *Match whole word only*, *Ignore “&”* and *Beginning with* search modifiers) produces no or few results.

- ***Set temporary markers* Checkbox**
  Check this option to set a temporary (blue dot) marker (●) next to each found item.

- ***Remove current markers* Checkbox**
  Check this option to remove all existing temporary (blue dot) markers (●) before setting new ones (in the subsequent find operation).

- ***Max. found results* Field**
  Enter the maximum number of results that can be returned by the find operation here.

- ***Find where* Tab Pages**
  Specifiy where to search for the specified character or text: in the item IDs, the source language, and/or the target language. Select the *Current View* tab to search in the active view. Use the *Project* tab to search in the active project regardless of the text that is currently visible in RC-WinTrans and/or

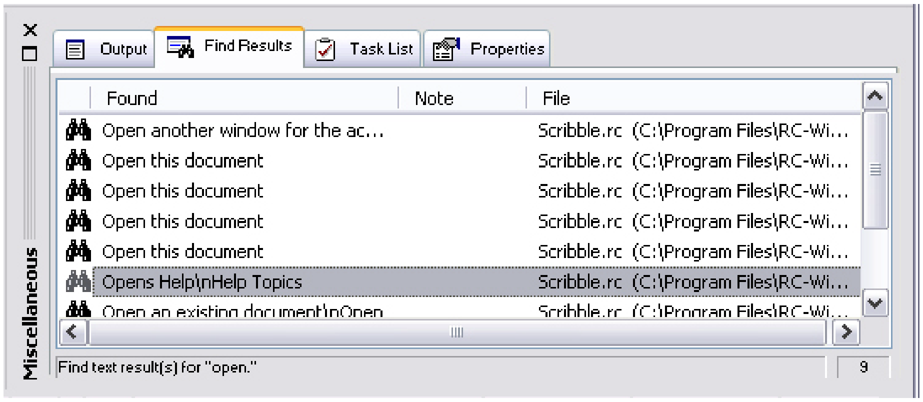
to search in other (open or closed) projects.

- ***Find*** **Button**
  Starts the find operation and closes the dialog box.

Results of the Find Operation

The results of a find operation will be displayed in *the Find Results* tab view in the *Miscellaneous* bar (Figure 2). Double-click on a list entry to select the corresponding translation unit in the *Text Table* view/*Translation Edit* bar.

Figure 2: The *Find Results* tab in the *Miscellaneous* bar.



Found	Note	File
Open another window for the ac...		Scribble.rc (C:\Program Files\RC-Wi...
Open this document		Scribble.rc (C:\Program Files\RC-Wi...
Open this document		Scribble.rc (C:\Program Files\RC-Wi...
Open this document		Scribble.rc (C:\Program Files\RC-Wi...
Open this document		Scribble.rc (C:\Program Files\RC-Wi...
Opens Help\nHelp Topics		Scribble.rc (C:\Program Files\RC-Wi...
Open an existing document\nOpen		Scribble.rc (C:\Program Files\RC-Wi...

Find text result(s) for "open."   9

Since all the items matching the search criteria get a temporary marker, they can easily be isolated from the other data by clicking on the *Show Marked Items Only* toolbar button in the *Project* window.

3.5.8 Translating Repetitions

RC-WinTrans lets you translate all items with the same source text (repetitions) simultaneously so you don't have to spend time entering in the same translation over and over again. The translation text is automatically applied to all repetitions when you confirm it with the *OK (__) (Translate All Repetitions)* button in the *Translation Edit* bar. There is also a feature to allow individual items to be translated uniquely if this is required.

***OK (__)* "Counter" Button (*Translation Edit* Bar)**

The total number of items having the source text of the selected item is indicated in the parentheses shown on the *OK (__) (Translate All Repetitions)* button in the *Translation Edit* bar. Clicking on this button after entering a translation text accepts the translation for all items with the same source text.

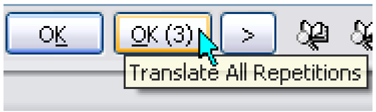
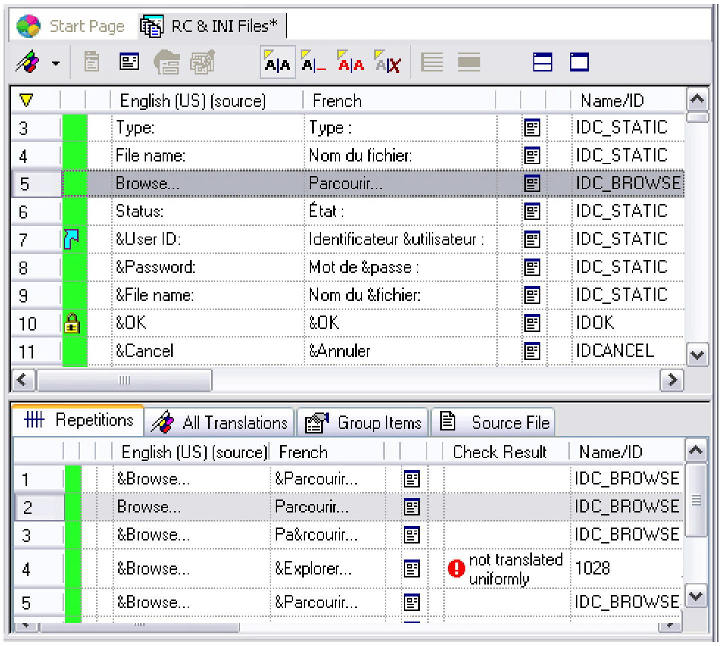


Figure 1: The *OK (__) (Translate All Repetitions)* button in the *Translation Edit* bar, here indicating a total of three occurrences of the selected source text.

***Repetitions* View (*Project* Window)**

The *Repetitions* view in the *Project* window lists all of the repetitions for the item currently selected in the *Text Table* view. All translation units that have the exact same source text as the selected item are considered repetitions; the view shows all the repetitions contained in the entire translation project (all project files).

Figure 2: A *Project* window with the *Text Table* view on top and the *Repetitions* view active in the secondary view below it.



		English (US) (source)	French	Name/ID
3		Type:	Type :	IDC_STATIC
4		File name:	Nom du fichier:	IDC_STATIC
5		Browse...	Parcourir...	IDC_BROWSE
6		Status:	État :	IDC_STATIC
7		&User ID:	Identificateur &utilisateur :	IDC_STATIC
8		&Password:	Mot de &passe :	IDC_STATIC
9		&File name:	Nom du &fichier:	IDC_STATIC
10	🔒	&OK	&OK	IDOK
11		&Cancel	&Annuler	IDCANCEL

	English (US) (source)	French	Check Result	Name/ID
1	&Browse...	&Parcourir...		IDC_BROWSE
2	Browse...	Parcourir...		IDC_BROWSE
3	&Browse...	Pa&rcourir...		IDC_BROWSE
4	&Browse...	&Explorer...	not translated uniformly	1028
5	&Browse...	&Parcourir...		IDC_BROWSE

Repetitions with Unique Translations

An item with a source text that is identical to other source texts in the project is **not** automatically translated under the following circumstances:

OK Pressing the *OK* button will accept the translation for the current item only.

🔒 Simultaneous translation of repetitions will not be applied to an item bearing the *Locked* symbol.

✕ Simultaneous translation of repetitions will not be applied to items that have been excluded from translation.

1x Simultaneous translation of repetitions will not be applied to an item bearing the *Unique Translation* symbol.

3.5.9 Automatic Translation

You can save yourself time and translation costs by using RC-WinTrans' auto-translate feature to automatically translate one or more text items based on translation data available from various sources. The auto-translate function can search the current project, dictionaries/translation memories, and/or other opened projects for translations that meet a predefined minimum match value and apply them to translation units in the active project.

The auto-translate function can be performed for all or a selection of the data contained in a translation project:

- **Auto-Translate All**
  Runs the auto-translate operation for all the text items in the active translation project. Use the *Auto-Translate All Text Items* command from the *Project* menu.
- **Auto-Translate Selected Items**
  Runs the auto-translate operation for those text items which are currently selected in the *Text Table* view/*Translation Edit* bar. Use any of the following:

  1. *Dictionaries* menu | *Auto-Translate* command.
  2. The *Auto-Translate* button (🔍) in the *Dictionaries* toolbar.
  3. The *Auto-Translate* button (🔍) in the *Translation Edit* bar.
  4. Shortcut: Shift+F4.

Data Sources for Automatic Translation

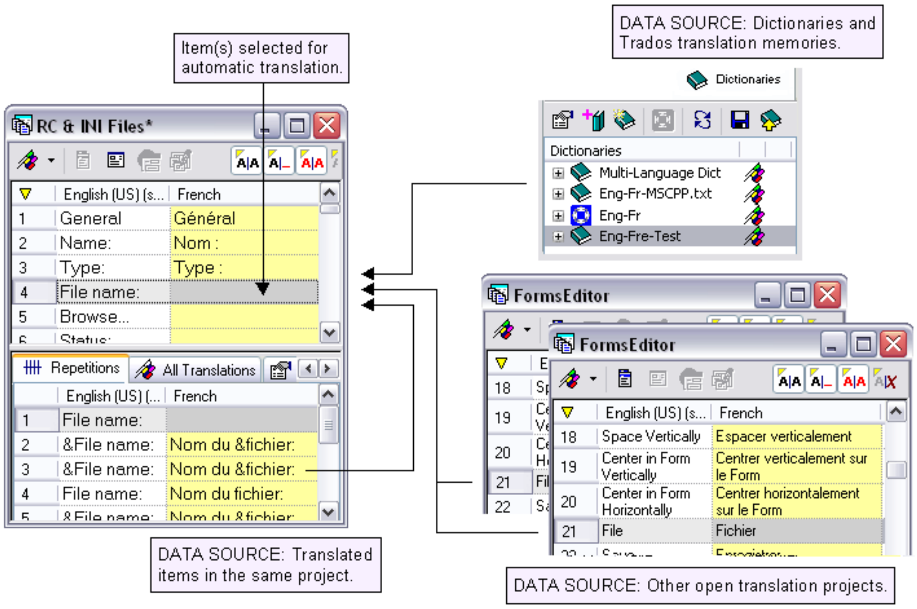
In order to translate a text item (a translation unit), the automatic translation mechanism looks up translations that already exist in the active project and/or in other available data sources, i.e., dictionaries, Trados™ translation memories, and other open translation projects.

Data sources used for automatic translation:

- Dictionaries/TMs that have been designated as "find references" in the *Workspace* bar's *Dictionaries* tab view (see Section 3.6.4.1 for more info).
- All other open translation projects.
- Existing translations in the active project.

Data sources can be specified on the *Automatic Translation* tab in the *Dictionary/TM Options* dialog box (Figure 2).

Figure 1: Data sources for RC-WinTrans' auto-translate function.



Item(s) selected for automatic translation.

DATA SOURCE: Dictionaries and Trados translation memories.

	English (US) (s...	French
1	General	Général
2	Name:	Nom :
3	Type:	Type :
4	File name:	
5	Browse...	
6	Status:	

	English (US) (...	French
1	File name:	
2	&File name:	Nom du &fichier:
3	&File name:	Nom du &fichier:
4	File name:	Nom du fichier:
5	&File name:	Nom du &fichier:

Dictionaries
Multi-Language Dict
Eng-Fr-MSCPP.txt
Eng-Fr
Eng-Fre-Test

	English (US) (s...	French
18	Space Vertically	Espacer verticalement
19	Center in Form Vertically	Centrer verticalement sur le Form
20	Center in Form Horizontally	Centrer horizontalement sur le Form
21	File	Fichier

DATA SOURCE: Translated items in the same project.

DATA SOURCE: Other open translation projects.

Options: *Automatic Translation* Tab Page

The *Automatic Translation* tab in the *Dictionary Options* dialog box contains various settings for the auto-translate mechanism and has a list of the data sources which are currently available to automatically translate data in the active source and target languages (i.e., only data sources that have same source and target languages as the ones for the currently active project can be used).

Open the *Dictionary/TM Options* dialog box by clicking on the *Dictionary Options* button in the *Dictionaries* toolbar (shown left) or by using the *Find and Get Translation Options* command from the *Dictionaries* menu.

- ***Data Source(s) (Search in)* Groupbox**

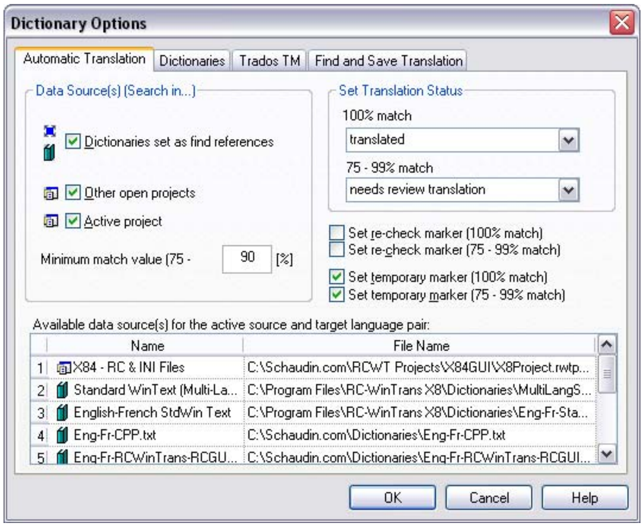
  Use the three checkboxes to specifiy which source(s) you want used in the auto-translate operation. RC-WinTrans will search for matching translations in the active project, in other open projects, and/or in dictionaries, depending on which options are checked here. Use the edit field to enter in a minimum match value that specifies how similar the two (source) texts must be before auto-translation can occur. The range may lie anywhere between 40% and 100%.

- ***Data Source(s) (Search in)*** **Groupbox**
  Use the three checkboxes to specifiy which source(s) you want used in the auto-translate operation. RC-WinTrans will search for matching translations in the active project, in other open projects, and/or in dictionaries, depending on which options are checked here. Use the edit field to enter in a minimum match value that specifies how similar the two (source) texts must be before auto-translation can occur. The range may lie anywhere between 40% and 100%.

- ***Set Translation Status*** **Groupbox**
  Assign a certain translation status to items that were automatically translated so you can easily find and review them in the project. Select one state for texts that matched 100% and select another for texts that had a match value of less than 100%.

- ***Set re-check marker / temporary marker*** **Checkboxes**
  Another way of marking items that were automatically translated is to set a re-check marker or a temporary (blue dot) marker (●) for them. Use the checkboxes given here to set your preferences.

- ***Available data sources*** **Display**
  Displays those sources which are currently available for use with the auto-translate function. It lists data sources that support the same source and target language as in the active translation project.



Data Source(s) (Search in...)

☑ Dictionaries set as find references

☑ Other open projects

☑ Active project

Minimum match value (75 - 90 [%]

Set Translation Status

100% match: translated

75 - 99% match: needs review translation

☐ Set re-check marker (100% match)
☐ Set re-check marker (75 - 99% match)
☑ Set temporary marker (100% match)
☑ Set temporary marker (75 - 99% match)

Available data source(s) for the active source and target language pair:

	Name	File Name
1	X84 - RC & INI Files	C:\Schaudin.com\RCWT Projects\X84GUI\X8Project.rwtp...
2	Standard WinText (Multi-La...	C:\Program Files\RC-WinTrans X8\Dictionaries\MultiLangS...
3	English-French StdWin Text	C:\Program Files\RC-WinTrans X8\Dictionaries\Eng-Fr-Sta...
4	Eng-Fr-CPP.txt	C:\Schaudin.com\Dictionaries\Eng-Fr-CPP.txt
5	Eng-Fr-RCWinTrans-RCGU...	C:\Schaudin.com\Dictionaries\Eng-Fr-RCWinTrans-RCGUI...

Figure 2: The *Dictionary Options* dialog box with the *Automatic Translation* tab active.

Assigning a Translation Status to Automatically Translated Items

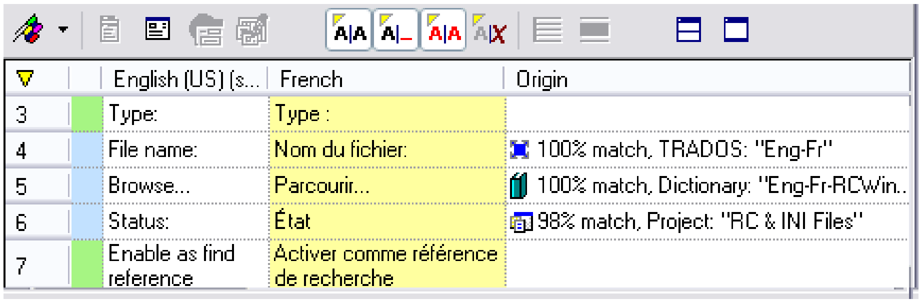
The auto-translate function can assign a specific translation status to items that are translated automatically, making it easier to locate and review them in the project. You can specify one state for items that were translated with a match value of 100% and another for items that had a match value of less than 100%. Translation states include “needs review translation,” “translated,” and “final.” The settings for the translation status are located on the *Automatic Translation* page in the *Dictionary/TM Options* dialog box (see previous section).

Designating a translation status for automatically translated items is recommended, especially for texts with a match value of less than 100%, which you will want to check for accuracy. Even translations from 100%-matching source texts should be reviewed to verify that they fit the context. Specified states let you find (and filter/isolate) these texts very easily.

Tracking the Origin of a Translation

RC-WinTrans maintains information (name and type) about the data source from which a translation was taken during the auto-translate operation. This information is displayed in the *Text Table* view’s “Origin” column.

Figure 3 shows the *Project* window’s *Text Table* view in which three items (lines 4, 5 and 6) have been translated automatically. The “Origin” column shows that each item was translated using a different data source (a Trados translation memory (4), a dictionary (5), and another translation project (6)). Note that the item in line 6 (“Status”) was only translated with a match value of 98% while the other two items were translated with source texts that matched 100%.



▽		English (US) (s...	French	Origin
3	🟩	Type:	Type :	
4	🟦	File name:	Nom du fichier:	100% match, TRADOS: "Eng-Fr"
5	🟦	Browse...	Parcourir...	100% match, Dictionary: "Eng-Fr-RCWin..
6	🟦	Status:	État	98% match, Project: "RC & INI Files"
7	🟩	Enable as find reference	Activer comme référence de recherche	

Figure 3: Automatically translated items in the *Text Table* view.

NOTE: The *Text Table* view's "Origin" column is set to be hidden by default. Make the column visible in the *Show/Hide Columns* dialog box, opened by right mouse-clicking in the view to call up a local menu and selecting the *Show/Hide Columns* command provided there.

IMPORTANT NOTES ON AUTOMATIC TRANSLATION

- In order to use a dictionary as a source for the automatic translation function it must be enabled as a find reference (see Section 3.6.4.1) in the *Workspace* bar's *Dictionaries* tab view and it must support the same source and target language combination that is currently active for the project.
- In order to use another translation project as a source for the automatic translation function it must currently be open in RC-WinTrans and the same source and target languages must be active.
- The auto-translate function **will not** be performed on items that have the translation status "translated" or "final"- these items will be skipped.

Undo Automatic Translation

You can undo the last auto-translate operation (delete all translations entered in this way) with the *Undo* command from the *Edit* menu, or with the "Ctrl+Z" shortcut.

Alternatives to Automatic Translation

1. **Get Translation(s) from Dictionary**
   Translations can also be imported directly from a selected dictionary using the *Get Translation(s)* command from the *Dictionaries* menu.

2. **Find a Translation for a Single Selected Item**
   To translate a single item manually you can use the find translation function which searches for translations in dictionaries (that have been set as find references. Unlike the auto-translate function, the *Find Translation* command only lists the translation(s) it has found (in the *Translation Edit* bar's *Found Results* list)- it does not automatically enter or save a translation.

3.6 Dictionaries

RC-WinTrans features extended support for the translation process with the use of dictionaries (in TMX or TXT file format) and (third-party) translation memories (e.g., SDL TRADOS™). These resources can be used to find translations for one, some, or all items needing to be translated. Depending on the number of resources available and their contents, this feature can significantly reduce the number of items needing to be translated manually by a translator, thereby reducing the overall cost of translation.

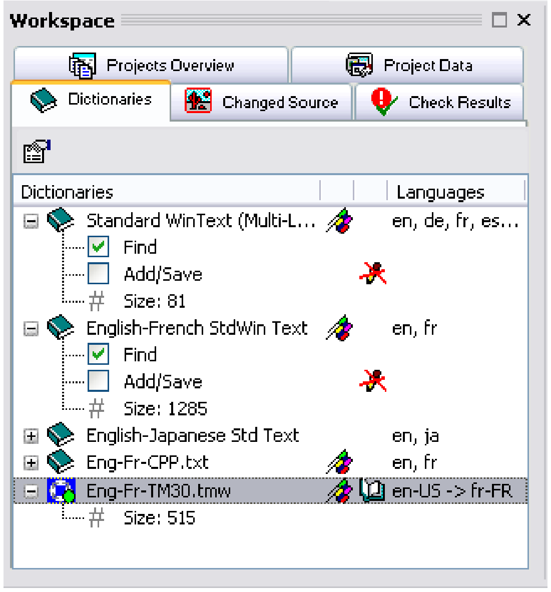
Dictionaries are typically used for the following tasks:

- **Finding** a translation for a text.
- **Automatic translation** of text items (find and get).
- **Saving** a translation to a dictionary for later use.
- **Importing/Exporting** translations for a number of selected text resources.

3.6.1 *Dictionaries* Tab View

The *Dictionaries* tab in the *Workspace* bar lists all the available dictionaries along with size, status, and access information for each one. Dictionaries can be added and removed using the *Dictionaries* menu.

In addition to text file-based dictionaries, the *Dictionaries* view can also contain **SDL TRADOS translation memories (TMs)**. RC-WinTrans works together with the Trados Translator's Workbench to access a Trados TM (file exten-sion: .tmw) just like a dictionary. The TM is listed in the *Dictionaries* view when open in the Translator's Workbench. See Section 3.7 of this guide for a description of the Trados integration in RC-WinTrans.



Dictionaries	Languages
Standard WinText (Multi-L...	en, de, fr, es...
☑ Find	
□ Add/Save	
# Size: 81	
English-French StdWin Text	en, fr
☑ Find	
□ Add/Save	
# Size: 1285	
English-Japanese Std Text	en, ja
Eng-Fr-CPP.txt	en, fr
Eng-Fr-TM30.tmw	en-US -> fr-FR
# Size: 515	

Figure: The *Dictionaries* view showing two (text file) diction-aries and a Trados translation memory available for use.

Status and Access Information (Symbols)

Closed Dictionary symbol. The dictionary is currently not in use.

Dictionary in Use symbol. The dictionary is open and in use. This symbol appears after a dictionary supporting the languages of the translation project is used in a find or a save text operation.

Trados Translation Memory symbol.

Language Support symbol. The dictionary supports the current source and target languages of the current translation project and can be used to find and save translation text.

Read-Only symbol. The dictionary is read-only. Translation text cannot be added/saved to the dictionary.

Checkboxes

The *Find* and *Add/Save* checkboxes in the *Dictionaries* view are used to include dictionaries in or exclude them from the find translation and save translation functions. When the *Find* checkbox is checked, the dictionary will be used to find translations (find, auto-find, auto-translate functions). The dictionary is then known as a **find reference**. When the *Add/Save* checkbox is checked, translations will be added to the dictionary (save, auto-save functions). The dictionary is then known as a **save reference**.

Size Information
The size of a dictionary (i.e., the number of entries it contains) is shown just below its *Find* and *Add/Save* checkboxes (see figure).

3.6.2 Dictionary Types

RC-WinTrans supports the following dictionary types (file types):

- Text format file dictionaries (.txt, .csv).
- TMX format file dictionaries (.tmx).
- SDL TRADOS™ translation memories (.tmw).

The RC-WinTrans system is designed to support the integration of other dictionary types as well (using VBA macros). Some developing skills are required- custom support can be purchased from Schaudin.com to implement this and other solutions.

3.6.2.1 Text File Dictionaries

Simple text files can be used as dictionaries given that the content follows a set of given rules, i.e., the first line is used to specify the dictionary's languages and the subsequent lines contain the dictionary data. RC-WinTrans comes with five text file dictionaries containing standard texts used in Windows software. These dictionaries are added to the *Dictionaries* view by the the setup program and is available when RC-WinTrans is started for the first time.

Additional information about the dictionaries supplied with RC-WinTrans is given below:

FILE NAME	LANGUAGES (Languge Code)
"Eng-Fr-MSCPP.txt"	"en"; "fr"
"Eng-Fr-Standard.txt"	"en"; "fr"
"En-Ja-Standard.txt"	"en"; "ja"
"En-Ja-VC60.txt"	"en"; "ja"
"MultiLangStdWinDict.txt"	"en"; "de"; "fr"; "es"; "it"; "nl"; "da"; "pt"; "sv"; "no"; "fi"; "pl"; "ru"; "cs"; "hu"; "he"; "ja"; "ko"; "th"; "el"; "tr"; "ar"; "zh-tw"; "zh-cn"

The dictionaries supplied with RC-WinTrans are located in the following folder of the RC-WinTrans program installation:

“<install dir>\RC-WinTrans X8\Dictionaries\”

File Encoding and Data Format

The character encoding of a text file used as a dictionary must be either ANSI or Unicode UTF-16-LE. The first line of the text file must contain a tab-separated list of the languages used in the file. The languages are coded according to ISO-639. You may use either the short two-letter form, e.g., “en,” or a combination of the language and country code, e.g., “en-us.”

Each subsequent line in the file holds a tab-separated list of dictionary items. The language order must be in the same order as the first line (languages specification) of the file. Each entry in the file must be written in quotation marks. This is meant to enhance the software to also work with text files that only use the tab character.

Example:

The first lines of a sample text dictionary combining two languages are shown below. The languages are specified in the first line: “en-uk” for English (UK) and “fr” for French:

“en-uk”	**-**	**“fr”**
“Abort”		“Abandonner”
“About”		“À propos de”
“About...”		“À propos de...”
“Absolute Position”		“Position absolue”

Source and Target Languages

Languages do not need to be stored in any particular order- RC-WinTrans only verifies the existence of a desired language pair. Languages are used by RC-WinTrans as the source and target language in any combination depending on the source and target language of the current project. The source and target languages are selected automatically.

Adding a Translation Unit

When saving or exporting translations to an existing text file dictionary, translations will be appended to the file. However, if RC-WinTrans finds that the file already contains a line with exactly the same combination of texts, that translation will not be exported. This is done to avoid double entries.

3.6.2.2 TMX File Dictionaries

RC-WinTrans supports the use of TMX (**t**ranslation **m**emory e**x**change) files combining two or more languages as dictionaries.

To avoid problems, TMX files are validated according to the TMX version that is used. RC-WinTrans comes with a set of XML schema files that allow for validating TMX Versions 1.1 through 1.4b.

NOTES:

1. RC-WinTrans does not handle any of the information stored in the TMX files other than the <seg> elements holding the translation-relevant data. In particular, meta-information as string category information is completely ignored.
2. If translation data is saved to an existing TMX file, the file will always be created anew. RC-WinTrans will only re-write the data from the TMX file that it has stored before, i.e., any data other than the <seg> elements will be lost. If you know that your TMX files hold other information, you should use a file copy for the dictionary.

3.6.2.3 Trados Translation Memories

RC-WinTrans supports the use of SDL TRADOS™ translation memories (file extension: .tmw) as registered dictionaries by interacting with the Trados Translator's Workbench (as an OLE automation server).

A Trados TM is displayed and handled within RC-WinTrans like any other (text file or TMX) dictionary except that there is a separate menu and toolbar for working with Trados.

See Section 3.7, “Trados and RC-WinTrans,” for further information.

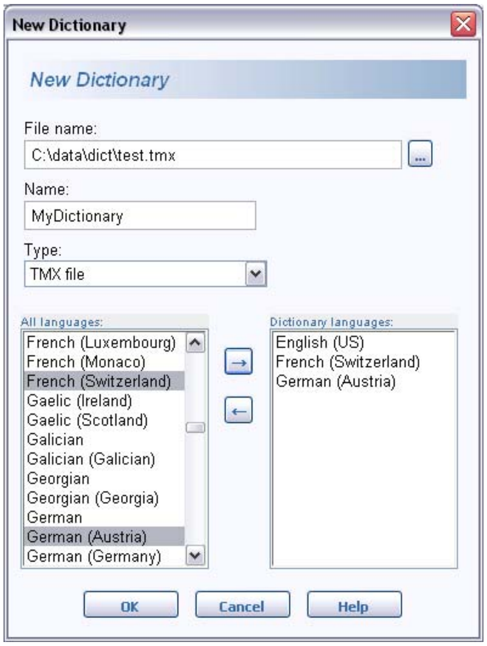
3.6.3 Adding New or Existing Dictionaries

In order to use a text or a TMX dictionary in RC-WinTrans, it must first be made accessible (registered) by being added to the *Dictionaries* view in the *Workspace* bar. You have the option of creating a brand new dictionary or of using an existing one.

RC-WinTrans supports two types (file types) of dictionaries: TMX files (file extension: .tmx) and text files (file extension: .txt).

Creating a New Dictionary

To create a new (empty) dictionary, use the *Add Dictionary* command from the *Dictionaries* menu. This command opens the *New Dictionary* dialog box (Figure 1).



File name: C:\data\dict\test.tmx

Name: MyDictionary

Type: TMX file

All languages:
French (Luxembourg)
French (Monaco)
French (Switzerland)
Gaelic (Ireland)
Gaelic (Scotland)
Galician
Galician (Galician)
Georgian
Georgian (Georgia)
German
German (Austria)
German (Germany)

Dictionary languages:
English (US)
French (Switzerland)
German (Austria)

Figure 1: *The New Dictionary* dialog box.

Next do the following in the *New Dictionary* dialog box:

1. Specify the name and the location of the new dictionary in the *File name* field. Use the *Browse* button to the right of the field to help determine the location.
2. (*Optional*) Enter a short name for the new dictionary in the *Name* field. This is the display name shown in RC-WinTrans' *Dictionaries* view.
3. Select the type of dictionary to create (text file or TMX file) from the *Type* combo-box.
4. Select two languages for the dictionary from the language combo-boxes provided. If desired, enable the dictionary as a "find reference" or "save reference" by clicking on the corresponding checkbox.
5. Press the *OK* button to create the new dictionary and add it to the *Dictionaries* tab view.

Using an Existing Dictionary

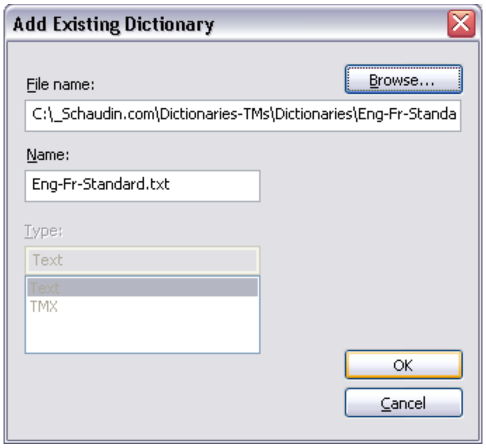
An existing dictionary (text or TMX file) can be added to the *Dictionaries* view in one of two ways:

- move the dictionary file from the File Explorer to the *Dictionaries* view using drag-and-drop, **OR**
- select the *Add Dictionary* command from the *Dictionaries* menu.

The second option (*Add Dictionary* command) opens the *Add Existing Dictionary* dialog box (Figure 2) where you can browse for/name the dictionary that you want to add to/use in RC-WinTrans. If desired, enable the dictionary as a “find reference” or “save reference” by clicking on the corresponding checkbox.

NOTE: Dictionaries that have been added to the *Dictionaries* view are enabled as find references (i.e., the *Find* checkbox for the dictionary is checked) by default. This means that the dictionary is automatically referenced in the find and auto-find operations, and, if desired, in the auto-translate operation as well (see Section 5.9 “Automatic Translation”). Conversely, dictionaries are **not** enabled as save references by default (i.e., the *Add/Save* checkbox is unchecked), which means that translations will **not** be saved to the dictionary when the *Save Translation* command is executed.

See Sections 3.6.4 through 3.6.4.2 for more on find and save references.



Add Existing Dictionary

File name: C:_Schaudin.com\Dictionaries-TMs\Dictionaries\Eng-Fr-Standa

Name: Eng-Fr-Standard.txt

Type: Text

Figure 2: The *Add Existing Dictionary* dialog box for registering a dictionary with RC-WinTrans.

Confirm your selections and close the *Add Existing Dictionary* dialog box by pressing the *OK* button. The dictionary will then be added to the *Dictionaries* view.

3.6.4 Using Dictionaries in RC-WinTrans

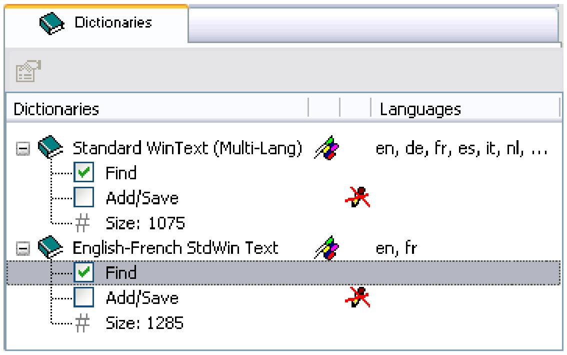
Once a dictionary has been made available to (registered with) RC-WinTrans (see previous section) it can be used to do the following:

- **find** a translation for a specific resource text.
- **save** a translation to a dictionary for later use.
- **automatically translate** text items.
- **import** translations for a number of selected text resources.
- **export** translations for a number of selected text resources.

A dictionary must be enabled as a **find reference** in order to use it for finding translation texts. Likewise, a dictionary must be enabled as a **save reference** in order to be able to save translations to it. This can be done directly in the *Dictionaries* view by clicking on the appropriate checkbox listed under each dictionary. Find and save references are described in greater detail in the next two sections.

3.6.4.1 Find References

Designating a dictionary as a find reference means that it will be referenced when searching for existing translations by RC-WinTrans' find, auto-find, and/or auto-translate functions. Dictionaries are enabled as find references by default. To activate/deactivate a dictionary as a find reference, check the *Find* checkbox below the dictionary's entry in the *Dictionaries* tab view.



Dictionaries

Dictionaries	Languages
⊟ Standard WinText (Multi-Lang)	en, de, fr, es, it, nl, ...
☑ Find	
□ Add/Save	
# Size: 1075	
⊟ English-French StdWin Text	en, fr
☑ Find	
□ Add/Save	
# Size: 1285	

Figure: The *Dictionaries* view with the *Find* checkboxes (for designating a dictionary as a find reference) checked.

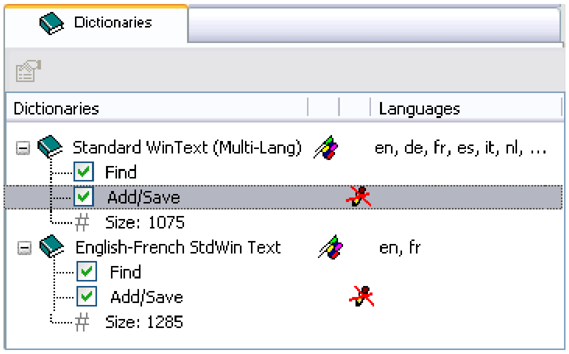
See Sections 3.5.9, 3.6.4.4, and 3.6.4.8 for descriptions of functions that use find references.

3.6.4.2 Save References

Designating a dictionary as a save reference means that translation data (source and target texts) will be added to it whenever the save translation or auto-save functions are used. Dictionaries are **not** enabled as save references by default. To activate/deactivate a dictionary as a save reference, check the *Add/Save* checkbox for below the dictionary entry in the *Dictionaries* tab view.

See Sections 3.6.4.7 and 3.6.4.8 for descriptions of functions that use save references.

NOTE: RC-WinTrans stores the information about texts that have been added to a selected dictionary(ies) until the application is closed, or, alternatively, until the *Save* button in the *Dictionaries* view is used, at which time the (dictionary) file(s) are updated to include the new data.



Dictionaries

Dictionaries	Languages
⊟ Standard WinText (Multi-Lang)	en, de, fr, es, it, nl, ...
☑ Find	
☑ Add/Save	
# Size: 1075	
⊟ English-French StdWin Text	en, fr
☑ Find	
☑ Add/Save	
# Size: 1285	

Figure: The *Dictionaries* view with the *Add/Save* checkboxes (for designating a dictionary as a save reference) enabled.

3.6.4.3 *Dictionaries* Toolbar

The *Dictionaries* toolbar features buttons to quickly access all the most important dictionary functions/commands:

Dictionary (Find/Save) Options
Opens the *Dictionary/TM Options* dialog box where you can set various options for working with dictionaries.

Find Translation
Runs the find translation operation for the text shown or selected in the *Translation Edit* bar's *Source* field. RC-WinTrans will search for text in all dictionaries set as find references. Use the drop-down button on the right to enable or disable the auto-find function.

Auto-Translate
Automatically translates items selected in the *Text Table* view for which the *Target* field is empty (i.e., no target text/translation exists) using selected sources (dictionaries, open project(s)). Sources are specfied in the *Dictionary/TM Options* dialog box along with other automatic translation settings. See Section 3.5.9, "Automatic Translation," for more details.

Save Translation(s)
Saves the current source text and its translation (or multiple items if selected in the *Text Table* view) to dictionaries that have been designates as save references. Use the drop-down button on the right to enable or disable the auto-save function.

NOTE: The *Find Translation(s)*, *Auto-Translate*, and *Save Translation(s)* commands are also available in the *Translation Edit* bar.

3.6.4.4 Finding Translations

Search for translations for the current source text (shown in the *Translation* Edit bar's *Source* field) using one of the methods described below. RC-WinTrans will search in all dictionaries that have been designated as find references (see Section 3.6.4.1).

1. ***Dictionaries* toolbar or menu**
   Press the *Find Translation* toolbar button (shown left) or select the *Find Translation* command from the *Dictionaries* menu. The *Translation Edit* bar has a button for this command as well.

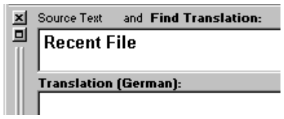
2. ***Source* Edit Field (*Translation Edit* bar)**
   Press the "Return/Enter" key while the *Source* edit field in the *Translation Edit* bar has the focus (input cursor).*

3. **F4 or Shift+F4 shortcuts**
   Use the F4 (find) function key to execute the *Find Translation* command, or use the "Shift+F4" key combination (auto-translate) to find a translation and automatically copy the best result to the *Translation Edit* bar's *Target* edit field.

* NOTE: RC-WinTrans will search for whatever is shown or selected in the *Source* text edit field when the find operation is executed. This means that you can enter a different text in this field (and press the "Return" key) and that is what will RC-WinTrans will look for. Select all or part of a text in the field and only texts matching the selection will be returned.

Example:

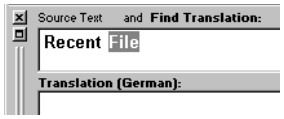
The source text "Recent File" is not selected. The *Find* command will be performed for the entire source item text ("Recent File").



Recent File

Example:

Only one word of the source text is selected. The *Find* and *Find and Get* commands will be performed for the selection ("File").

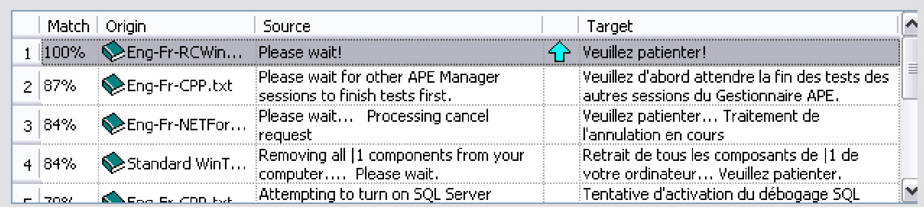


Recent File

Options for finding translations in dictionaries can be set on the *Dictionaries* tab in the *Dictionary/TM Options* dialog box (see Section 3.6.4.6 and its figure).

3.6.4.5 *Dictionary Found Results* List

The results of a find translation operation (see previous section) are listed in the *Dictionary Found Results list* (lower portion of the *Translation Edit* bar, see figure below). Click on the *Translation Edit* bar's *Insert Translation* symbol (⇧) to copy the translation text for the selected list item (in the *Dictionary Found Results* list) to the *Target* edit field.

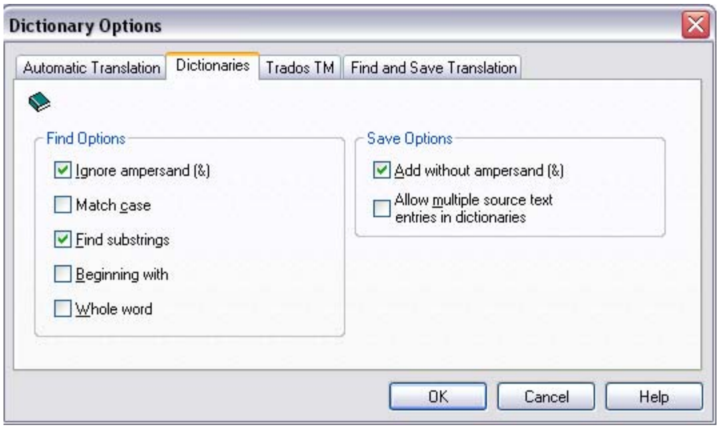


	Match	Origin	Source		Target
1	100%	Eng-Fr-RCWin...	Please wait!	⇧	Veuillez patienter!
2	87%	Eng-Fr-CPP.txt	Please wait for other APE Manager sessions to finish tests first.		Veuillez d'abord attendre la fin des tests des autres sessions du Gestionnaire APE.
3	84%	Eng-Fr-NETFor...	Please wait... Processing cancel request		Veuillez patienter... Traitement de l'annulation en cours
4	84%	Standard WinT...	Removing all |1 components from your computer.... Please wait.		Retrait de tous les composants de |1 de votre ordinateur... Veuillez patienter.
5	79%	Eng-Fr-CPP.txt	Attempting to turn on SQL Server		Tentative d'activation du débogage SQL

Figure: Matching translations found and listed in the *Translation Edit* bar's *Dictionary Found Results* list.

3.6.4.6 Dictionary Find and Save Options

The criteria for finding and saving translations in/to dictionaries can be specified in the *Dictionary/TM Options* dialog box. Open this dialog box
with the *Find and Get Translation Options* command from the *Dictionaries* menu, or use the *Find and Get Translation Options* button in the *Dictionaries* toolbar (above, left). Select the ***Dictionaries* tab** in the dialog (see figure below) to change the settings for finding/saving a translation(s) in/to dictionaries.



Dictionary Options

Find Options
- ☑ Ignore ampersand (&)
- □ Match case
- ☑ Find substrings
- □ Beginning with
- □ Whole word

Save Options
- ☑ Add without ampersand (&)
- □ Allow multiple source text entries in dictionaries

Figure: The *Dictionaries* tab in the *Dictionary/TM Options* dialog box.

3.6.4.7 Saving Translations in Dictionaries

Save the currently selected translation unit to a dictionary (that has been designated as a save reference) by clicking on the *Save Translation* toolbar button or by using the same command from the *Dictionaries* menu. Texts can only be saved to a dictionary that supports the project's current source and target languages (and that is enabled as a save reference-see Section 3.6.4.2).

Save several items to a dictionary all at once by selecting them in the *Text Table* view and then using either of the aforementioned commands. Alternatively, you might also consider using the export function, described in Section 3.6.4.9 of this guide.

Two options for saving texts in dictionaries set as save references can be set on the *Dictionaries* tab in the *Dictionary/TM Options* dialog box (see Section 3.6.4.6 and its figure).

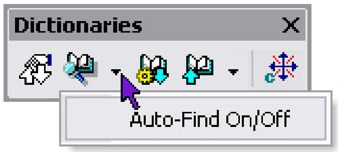
NOTE: RC-WinTrans stores the information about texts that have been added to a selected dictionary(ies) until the application is closed, or, alternatively, until the *Save* command in the *Dictionaries* view is used (local menu accessed via right mouse-click). The (dictionary) file is then updated to include the new data.

3.6.4.8 Auto-Find, Auto-Save, and Auto-Translate Functions

RC-WinTrans's automated functions for dictionaries and Trados translation memories make it even easier to translate text and build up dictionaries with the addition of that data. Please note that the auto-find and auto-save functions are automated in the true sense of the word (i.e., find/save will be performed automatically when activated), whereas the auto-translate function only runs on command (here "auto" refers to the means of translation rather than how the function is executed).

Descriptions of the auto-find, auto-save, and auto-translate functions follow.

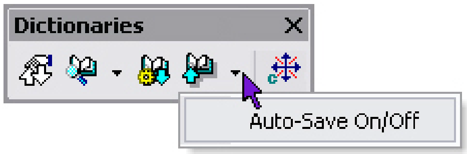
Auto-Find



Enabling this function tells RC-WinTrans to automatically search for a translation text whenever a (text-type) resource is selected for editing. RC-WinTrans will only search for translations in dictionaries that have been designated as find references (see Section 3.6.4.1).

Activate or deactivate the auto-find function by clicking the drop-down button on the right of the *Find Translation* button in the *Dictionaries* toolbar and selecting the *Auto-Find On/Off* option (shown above). This command is also available in the *Dictionaries* menu.

Auto-Save

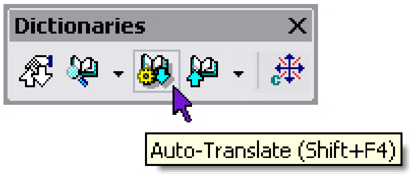


The auto-save function copies and saves the the current source text and its translation from the *Translation Edit* bar to all dictionaries that have been designated as save references (see Section 3.6.4.2). When auto-save is activated, the save function will be executed automatically when a translation is confirmed using the *OK* or *OK (__)* button in the *Translation Edit* bar or by pressing the Enter/Return key.

Acivate or deactivate auto-save by clicking the drop-down button on the right of the *Auto-Save* button in the *Dictionaries* toolbar and selecting the *Auto-Save On/Off* option. This command is also available in the *Dictionaries* menu.

NOTE: Translations are not (physically) saved to dictionaries until the project is closed or, alternatively, until the *Save* command in the *Dictionaries* view (local menu) is used to transfer the data to the appropriate (dictionary) file.

Auto-Translate



The auto-translate function uses selected data sources (dictionaries/TM, the active project, other open projects) to automatically translate texts according to a set of predefined match requirements. Run the auto-translate function for the currently selected item (*Text Table* view/*Translation Edit* bar) or for several selected items (*Text Table* view) by clicking on the *Auto-Translate* button (shown above). This command is also available in the *Dictionaries* menu. Another command for automatically translating all the text items in the active project is located in the *Project* menu (*Auto-Translate All Text Items* command).

The auto-translate function **will not** be performed on items that have the translation status “translated” or “final”- these items will be skipped.

Section 3.5.9 of this guide discusses the auto-translate function in greater detail including how to select the sources used in the operation, set options, and monitor results.

3.6.4.9 Importing and Exporting Translations

Import (*Get Translation(s)* Command)

Translation text can be imported from a dictionary for one or several items in a project with the *Get Translation(s) from Dictionary* command.

The term “import” refers to the process of searching for matching translations and inserting them as the target texts for selected items. A dictionary must support the project’s active source and target languages in order for it to be used in the import operation.

To import translations for one or more items in the current project, begin by selecting the desired entry or entries in the *Text Table* view, then press the *Import* button in the *Project* toolbar (shown above) and select the *Get Translation(s) from Dictionary* option (Figure 1). A menu command for this operation is also available in the *Dictionaries* menu.

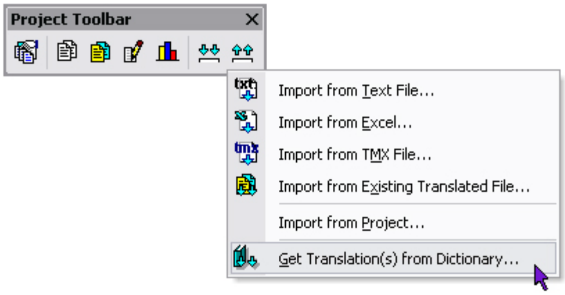
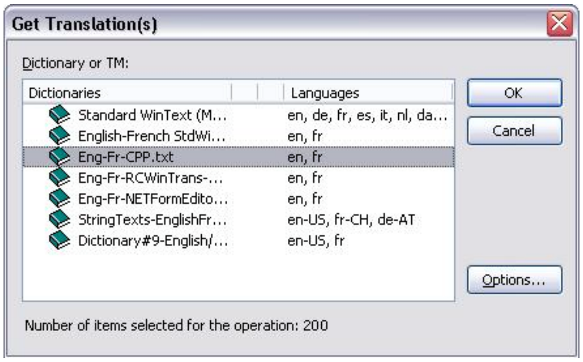


Figure 1: The *Get Translaion(s) from Dictionary* command accessed via the *Import* button in the *Project* toolbar (second button from the right).

Selecting this command opens the *Get Translation(s)* dialog (Figure 2) where the dictionaries available for the import operation are listed and settings can be defined (*Options* button). Confirm your selections and proceed by clicking on the *OK* button.



Dictionaries	Languages
Standard WinText (M...	en, de, fr, es, it, nl, da...
English-French StdWi...	en, fr
Eng-Fr-CPP.txt	en, fr
Eng-Fr-RCWinTrans-...	en, fr
Eng-Fr-NETFormEdito...	en, fr
StringTexts-EnglishFr...	en-US, fr-CH, de-AT
Dictionary#9-English/...	en-US, fr

Number of items selected for the operation: 200

Figure 2: The *Get Translation(s)* dialog box used to carry out and access the settings for the import of translation texts from a selected dictionary.

Pressing the *OK* button closes the dialog box and tells RC-WinTrans to start the import. Translations in the dictionary matching the given specifications will be imported for the selected item(s).

A blue dot (temporary) marker is placed beside each imported translation. You can use the *Next Marked Item* and *Previous Marked Item* toolbar buttons (shown right) to move between markers.

Export (*Add Translation(s)* Command)

Translation texts can also be exported (i.e., added) to the dictionary of your choice. A selected dictionary can only be used for the export operation if it supports the project's active source and target languages.

Use the *Add Translation(s)* command to export (add) one or more selected translation units from the current project to a dictionary. First, select the item(s) that you want to export in the *Text Table* view, then press the *Export* button in the *Project* toolbar (shown above) and select the *Add Translation(s) to Dictionary* option (Figure 3). A menu command for this operation is also available in the *Dictionaries* menu.

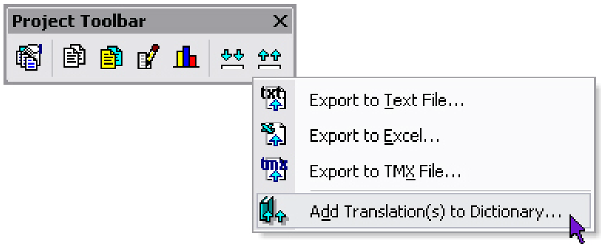
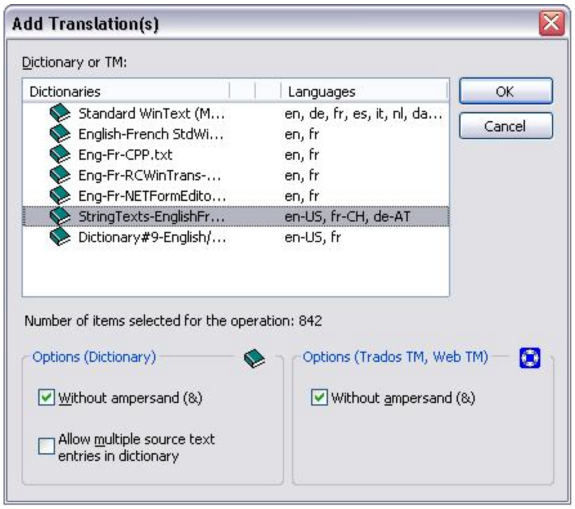


Figure 3: The *Add Translaion(s) to Dictionary* command accessed via the *Export* button (far right) in the *Project* toolbar.

The *Add Translation(s) to Dictionary* command will call up the *Add Translation(s)* dialog box (Figure 4) to run and/or define the settings for the export operation. Confirm your selections and proceed by clicking on the *OK* button.



Add Translation(s)

Dictionary or TM:

Dictionaries	Languages
Standard WinText (M...	en, de, fr, es, it, nl, da...
English-French StdWi...	en, fr
Eng-Fr-CPP.txt	en, fr
Eng-Fr-RCWinTrans-...	en, fr
Eng-Fr-NETFormEdito...	en, fr
StringTexts-EnglishFr...	en-US, fr-CH, de-AT
Dictionary#9-English/...	en-US, fr

Number of items selected for the operation: 842

Options (Dictionary)

☑ Without ampersand (&)

□ Allow multiple source text entries in dictionary

Options (Trados TM, Web TM)

☑ Without ampersand (&)

Figure 4: The *Add Translation(s)* dialog box used to run and specify the conditions for the export of translation texts to a selected dictionary.

NOTE: RC-WinTrans stores the information about texts that have been added to a selected dictionary(ies) until the application is closed, or, alternatively, until the *Save* command in the *Dictionaries* view (local menu) is used to update the (dictionary) file to include the new data.

3.7 Trados and RC-WinTrans

3.7.1 Getting Started

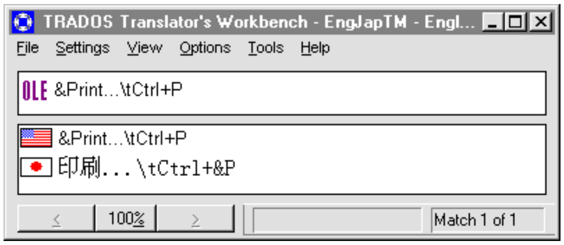
RC-WinTrans can be used in conjunction with the Trados™ Translator's Workbench. An integrated GUI (graphical user interface) component makes it possible for RC-WinTrans to interact with the Translator's Workbench. A Trados translation memory (TM) can be accessed from within RC-WinTrans to find, get, and save translations.

NOTE: Trados Versions 3 through 7.5 are supported. RC-WinTrans is not yet able to work with the new SDL Studio 2009.

When the Trados Translator's Workbench is installed on the current machine, RC-WinTrans adds a *Trados* menu to its menu bar and a *Trados* toolbar with commands relative to the Workbench and the active TM.

RC-WinTrans will start the Translator's Workbench on command (see the toolbar button below) to access a Trados TM.

The *Start/Close Trados Translator's Workbench* toolbar button.



OLE &Print...\tCtrl+P

&Print...\tCtrl+P

印刷...\tCtrl+&P

Match 1 of 1

Figure 1: The SDL TRADOS Translator's Workbench application window.

Once the Translator's Workbench is activated (started and running), RC-WinTrans automatically opens the last translation memory used. To open a TM, use the *Open TM* command in the *Trados* menu, or go to the Translator's Workbench window directly and use the "Open" command in the "File" menu.

To create a new translation memory, activate the Translator's Workbench window and select the "New" command from the "File" menu. RC-WinTrans does **not** have a command to create a new TM.

Functional Integration Summary

The integrated translation memory access supports several translation tasks within RC-WinTrans. These tasks are basic functions such as searching for translations, saving translations, and executing a concordance search, as well as extended operations such as the import (from Trados) and uniform translation check functions. Furthermore, TMs are integrated in and accessed by RC-WinTrans just like RC-WinTrans dictionaries.

The following functionality is available for Trados TMs:

- **Find (and Get) Translation(s)**
  Find and get a translation for a source text(s) needing to be translated.

- **Save Translation(s)**
  Save a translation(s) to a Trados TM.

- **Auto-Translate**
  Use the contents of a Trados TM to automatically translate one, more or all the texts in a translation project.

- **Concordance Search**
  Perform a concordance search in the active TM.

- **Auto-Find and Auto-Save**
  RC-WinTrans features automated functions for finding, getting, and saving translations from/to Trados translation memories. The actual translation/ modification of resources takes place in RC-WinTrans (*Translation Edit* bar). See Section 3.7.6 for additional information.

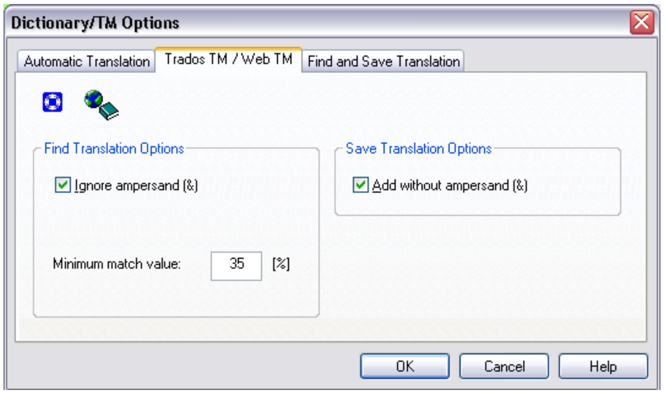
Opening/Closing a Translation Memory

Use the *Start/Close Trados Workbench* button (shown left) in the *Trados* toolbar to open/close the Translator's Workbench (open/close a TM).

TECHNICAL NOTE ON RC-WINTRANS' CONNECTION TO THE TRANSLATOR'S WORKBENCH: RC-WinTrans' use of a Trados TM consists of starting the Trados Translator's Workbench and directing it to open the respective translation memory file (*.tmw). RC-WinTrans interacts with the Translator's Workbench via OLE automation where RC-WinTrans uses the Translator's Workbench as an OLE automation server. Whenever a Trados TM is used the Translator's Workbench application is started and runs in a separate window alongside RC-WinTrans. The Translator's Workbench is opened and closed from within RC-WinTrans.

Options

Options for finding and saving texts in a Trados TM (e.g., ignore ampersand (&) character) are set on *Trados TM/Web TM* tab in the *Dictionary/TM Options* dialog box (Figure 2). Open the dialog box by clicking on the *Find and Get Translation Options* toolbar button (shown above, left) or by selecting the same comand from the *Dictionaries* menu.



Find Translation Options

☑ Ignore ampersand (&)

Minimum match value: 35 [%]

Save Translation Options

☑ Add without ampersand (&)

Figure 2: The *Trados TM* tab active in the *Dictionary/TM Options* dialog box.

3.7.2 *Trados* Toolbar and Menu

The *Trados* toolbar (Figure 1) and *Trados* menu (Figure 2) are available if the Trados™ Translator's Workbench is installed. Both the menu and the toolbar provide commands for working with Trados translation memories (TMs) in conjunction with RC-WinTrans.

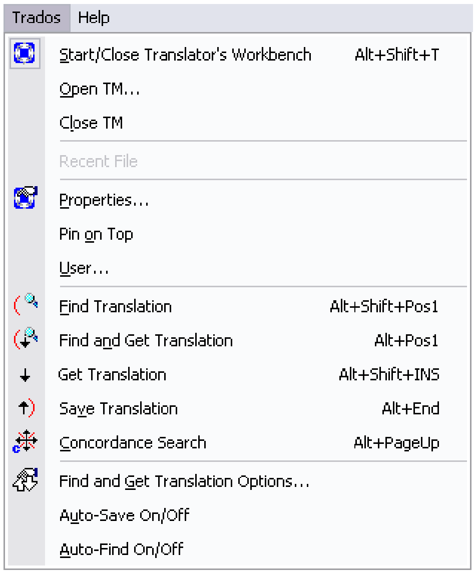
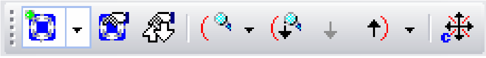

Trados
Help
Start/Close Translator's Workbench Alt+Shift+T
Open TM...
Close TM
Recent File
Properties...
Pin on Top
User...
Find Translation Alt+Shift+Pos1
Find and Get Translation Alt+Pos1
Get Translation Alt+Shift+INS
Save Translation Alt+End
Concordance Search Alt+PageUp
Find and Get Translation Options...
Auto-Save On/Off
Auto-Find On/Off

Figure 1: The *Trados* menu.

Figure 2: The *Trados* toolbar.



Commands

Start/Close Trados Translator's Workbench

Switch button. Starts and closes the Trados Translator's Workbench.

Status Indicators

If a translation memory is selected (i.e., active in the Translator's Workbench), a status indicator (red/green dot) appears on the toolbar button. A **red dot** indicates that the TM's language ID(s) does/do not match the one(s) for the current project's source and/or target language. When a red dot is shown the TM is disabled for all operations related to working on or with the TM. A **green dot** indicates that the translation project's source and target language IDs match those of the TM- the TM can be modified and used to support the translation process for that particular project.

Trados TM Properties

Opens the property page for the active TM. Moving the mouse cursor over the button displays a tooltip text indicating the source and target languages of the TM.

Trados: Find and Get Translation Options

Opens the *Dictionary/TM Options* dialog box where various options for working with dictionaries and translation memories are set.

Find Translation

Searches the active TM for a translation for the source text of the item currently selected in RC-WinTrans. Use the drop-down button on the right to enable/disable the auto-find function for the TM (see Section 3.7.6 for information about automated functions).

Find and Get (Auto-Translate)

Searches the active TM for a translation for the currently selected source item. The best matching translation is entered as the target text and saved. (NOTE: This function will **not** overwrite an existing translation.)

Get Translation

Copies the translation text selected in the Translator's Workbench to the *Target* edit field in RC-WinTrans' *Translation Edit* bar.

Save Translation
Saves the text in the *Translation Edit* bar's *Target* edit field to the active Trados TM. Use the drop-down button on the right to enable/ disable the auto-save TM (see Section 3.7.6 for information about automated functions).

Trados Concordance Search
Runs the concordance search for the item currently selected in RC-WinTrans.

3.7.3 Finding and Getting Translations

Find Operation
Search for translations for the current source text (shown in the *Translation* Edit bar's *Source* field) using one of the methods described below. RC-WinTrans will search in the active translation memory (open in the Translator's Workbench) as long as it is enabled as a "find reference," which is the default setting for an open TM.

1. *Trados* toolbar or menu
Press the *Find Translation* toolbar button (shown left) or select the *Find Translation* command from the *Trados* menu. The *Translation Edit* bar has a button for this command as well.

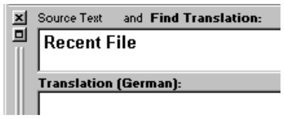
2. *Source* edit field (*Translation Edit* bar)
Press the Return/Enter key while the *Source* edit field in the *Translation Edit* bar has the focus (input cursor).*

3. F4 or Shift+F4 shortcuts (also: Alt+Shift+Pos1)
Use the F4 function key to execute the *Find Translation* command, or use the Shift+F4 key combination (auto-translate) to find a translation and automatically copy the best result to the *Translation Edit* bar's *Target* edit field.

* NOTE: RC-WinTrans will search for whatever is shown or selected in the *Source* edit field when the find operation is executed. This means that you can enter a different text in this field and that is what RC-WinTrans will search for. Select all or part of a text in the field and only texts matching the selection will be returned.

Example:

The source text “Recent File” is not selected. The *Find* command will be performed for the entire source item text (“Recent File”).



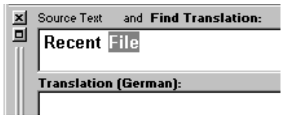
Source Text and **Find Translation:**

Recent File

Translation (German):

Example:

Only one word of the source text is selected. The *Find* and *Find and Get* commands will be performed for the selection (“File”).



Source Text and **Find Translation:**

Recent File

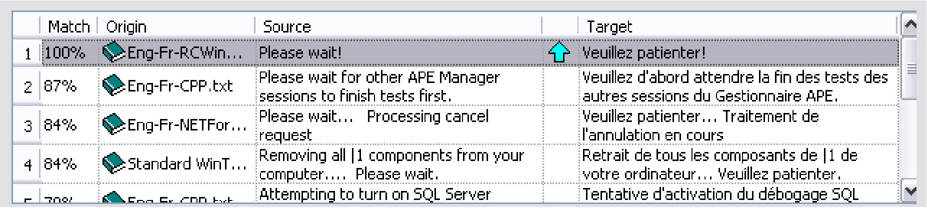
Translation (German):

The option for finding translations in Trados translation memories with or without the ampersand character (& – hotkey specification) can be set on the *Trados TM* property page in the *Dictionary/TM Options* dialog box (see Section 3.7.1 and its Figure 2).

“Get” (Insert/Use) Translation

The results of a find translation operation (see above) are listed in the *Dictionary Found Results list* (lower portion of the *Translation Edit* bar, see figure below). Use the *Translation Edit* bar’s *Get Found Result(s)* symbol (⇧) to copy the translation text for the selected list item (in the *Dictionary Found Results* list) to the *Target* edit field.

Figure: Matching translations found and listed in the *Translation Edit* bar’s *Dictionary Found Results* list.



	Match	Origin	Source		Target
1	100%	Eng-Fr-RCWin…	Please wait!	⇧	Veuillez patienter!
2	87%	Eng-Fr-CPP.txt	Please wait for other APE Manager sessions to finish tests first.		Veuillez d'abord attendre la fin des tests des autres sessions du Gestionnaire APE.
3	84%	Eng-Fr-NETFor…	Please wait… Processing cancel request		Veuillez patienter… Traitement de l'annulation en cours
4	84%	Standard WinT…	Removing all |1 components from your computer…. Please wait.		Retrait de tous les composants de |1 de votre ordinateur… Veuillez patienter.
5	79%	Eng-Fr-CPP.txt	Attempting to turn on SQL Server		Tentative d'activation du débogage SQL

To get translations for multiple items selected in the *Text Table* view, use the *Get Translation* command in the *Trados* menu, or consider using the auto-translate function (described in Section 3.5.9 of this guide).

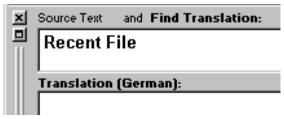
3.7.4 Concordance Search

RC-WinTrans can perform a concordance search for the active Trados translation memory (TM). The concordance search checks a text (shown or selected in the *Translation Edit* bar's *Source* field) against the data in the active TM. The command is available via the *Concordance Search* toolbar button (shown above) and in the *Trados* menu.

RC-WinTrans will search for whatever is shown or selected in the *Source* edit field when the find operation is executed. This means that you can enter a different text in this field and that is what will RC-WinTrans will look for. Select all or part of a text in the field and only texts matching the selection will be returned.

Example:

The source text "Recent File" is not selected. The *Find* command will be performed for the entire source item text ("Recent File").



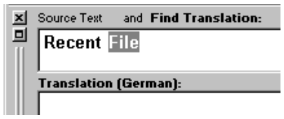
Source Text and Find Translation:

Recent File

Translation (German):

Example:

Only one word of the source text is selected. The *Find* and *Find and Get* commands will be performed for the selection ("File").



Source Text and Find Translation:

Recent File

Translation (German):

3.7.5 Saving Translations

The *Save* command adds the source and target text of the item currently selected in RC-WinTrans to the active Trados translation memory (TM) as a new translation unit (TU) or by overwriting an existing translation unit. It depends on whether the Trados TM is a single- or multiple-translations TM.

You can save a number of items to the the TM at once by selecting these in the *Text Table* view and then using the *Save* command. Only translated items are saved.

The option for saving translations in Trados translation memories with or without the ampersand character (& – hotkey specification) can be set on the *Trados TM* property page in the *Dictionary/TM Options* dialog box (see Section 3.7.1 and its Figure 2).

***Save Translation* (*Trados* Toolbar/Menu)**
The command to save a selected translation unit(s) to the active TM is located in the *Trados* menu and in the toolbar (*Save Translation*).

Auto-Save
Enable the auto-save function if you want translations that are edited in the *Translation Edit* bar to be saved to the active Trados TM automatically (when the translation is saved/accepted in RC-WinTrans). Use the drop-down button on the right part of the *Save Translation* button to enable/disable the auto-save function.

3.7.6 Automated Functions

RC-WinTrans performs two functions automatically whenever a text item is selected and/or translated in the *Translation Edit bar*. These are the auto-find and auto-save functions. Auto-find searches the active translation memory (TM) for translations for the selected source text while auto-save adds the translated item (source and target text) to the active TM.

Each automated function can be individually enabled and disabled in the *Trados* menu or in the drop-down menus for the *Find* and *Save* buttons in the *Trados* toolbar (see Figures 1 and 2 below).

Auto-Find

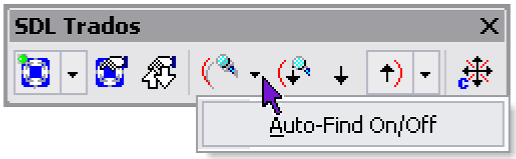
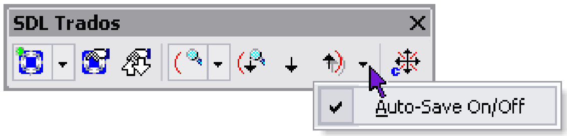


Figure 1: The *Find* toolbar button where the *Auto-Find On/Off* option has been selected in the button's drop-down menu.

Activate or deactivate the auto-find function by clicking on the drop-down button to the right of the *Find Translation* button in the *Trados* toolbar and selecting the *Auto-Find On/Off* option (shown above). This command is also available in the *Trados* menu.

Auto-Save

Figure 2: The *Save* toolbar button where *the Auto-Find On/Off* option has been selected in the button's drop-down menu.



Activate or deactivate auto-save by clicking on the drop-down button to the right of the *Auto-Save* button in the *Trados* toolbar and selecting the *Auto-Save On/Off* option. This command is also available in the *Trados* menu.

Automatic Translation (Find and Get)

An active Trados translation memory (TM) is automatically used (referenced) in an automatic translation operation if the TM has been enabled for this purpose on the *Automatic Translation* property page (*Dictionary/TM Options* dialog box).

The options for finding and obtaining translations in/from Trados TMs are easily accessed via the *Find and Get Translation Options* button in the Trados toolbar (Figure 1). The same command is also available in the *Trados* menu.

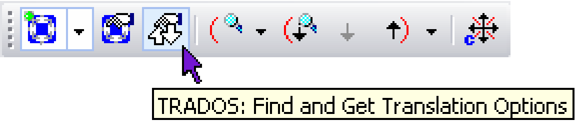
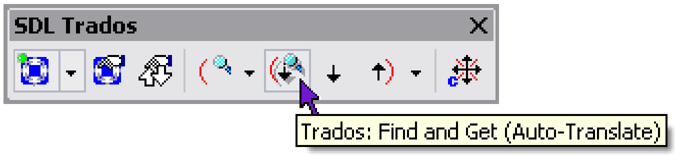


Figure 3: The *Find and Get Translation Options* button.

Figure 4: The *Find and Get (Auto-Translate)* button selected in the *Trados* toolbar.



The auto-translate function uses selected data sources (dictionaries/TMs, the active project, other open projects) to automatically translate texts according to a set of predefined match requirements. Run the auto-translate function for the currently selected item (*Text Table* view/*Translation Edit* bar) or for several selected items (*Text Table* view) by clicking on the *Find and Get (Auto-Translate)* button (shown above). This command is also available in the *Trados* menu. Another command for automatically translating all the text items in the active project is located in the *Project* menu (Figure 5).

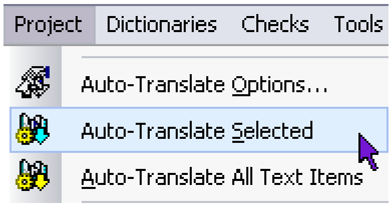


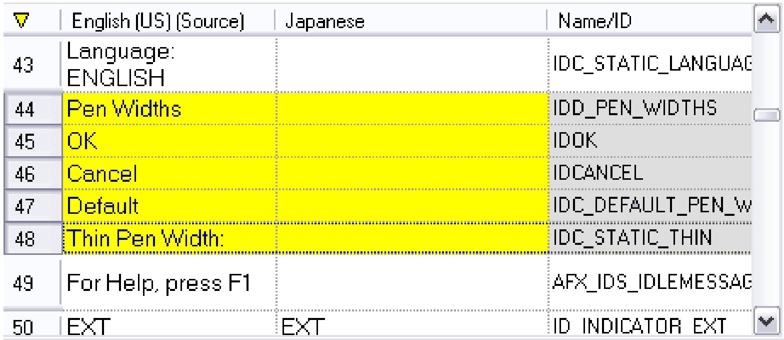
Figure 5: Automatic translation commands in the *Project* menu.

Section 3.5.9 of this guide discusses the auto-translate function in greater detail including how to select the sources used in the operation, set options, and monitor results.

3.7.7 Sample: Getting Translations from a Trados TM

The following is a step-by-step demonstration of how to get translations from a Trados translation memory (TM) for text entries in the current project's target language. This example shows you how to perform an import for a single item as well as for a number of items all at once. "Import" refers to the process of searching for matching translations and entering them as translations for selected items. A Trados TM must be accessible in order to perform this operation. The TM must also support the current project's source and target languages (in this example the source language is English and the target language is Japanese).

① To get translations for items in the current project, first select the entry or entries for which to import in the *Text Table* view.



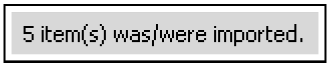
▽	English (US) (Source)	Japanese	Name/ID
43	Language: ENGLISH		IDC_STATIC_LANGUAG
44	Pen Widths		IDD_PEN_WIDTHS
45	OK		IDOK
46	Cancel		IDCANCEL
47	Default		IDC_DEFAULT_PEN_W
48	Thin Pen Width:		IDC_STATIC_THIN
49	For Help, press F1		AFX_IDS_IDLEMESSAG
50	EXT	EXT	ID_INDICATOR_EXT

Figure 1: Five non-translated items selected in RC-WinTrans' *Text Table* view.

② After selecting the items for which you want to get translations, use any of the following:

- the *Find and Get Translation* command in the *Trados* menu or the *Trados* toolbar, OR
- the *Auto-Translate* command in the *Project* menu.

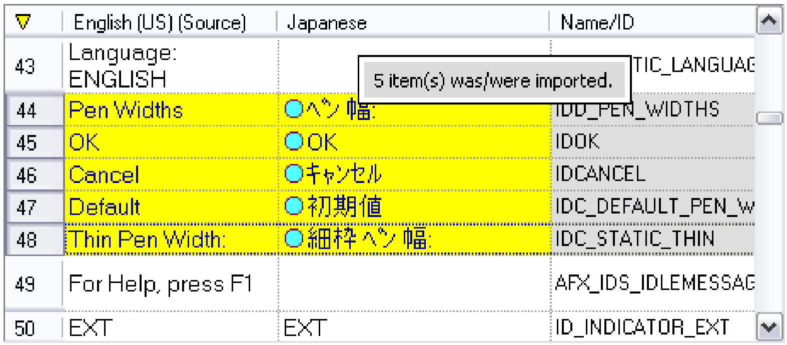
③ Start the import in RC-WinTrans. Translations in the TM matching the given specifications will be imported for the selected items. The number of text entries imported is displayed in a *Flash Info* window.



5 item(s) was/were imported.

Figure 2: The *Flash Info* window showing the number of imported items.

A blue dot (temporary) marker is placed beside each imported translation. You can use the *Next Marked Item* and *Previous Marked Item* toolbar buttons (shown right) to move between markers.



▽	English (US) (Source)	Japanese	Name/ID
43	Language: ENGLISH		TIC_LANGUAG
44	Pen Widths	●ペン幅:	IDD_PEN_WIDTHS
45	OK	●OK	IDOK
46	Cancel	●キャンセル	IDCANCEL
47	Default	●初期値	IDC_DEFAULT_PEN_W
48	Thin Pen Width:	●細枠ペン幅:	IDC_STATIC_THIN
49	For Help, press F1		AFX_IDS_IDLEMESSAG
50	EXT	EXT	ID_INDICATOR_EXT

5 item(s) was/were imported.

Figure 3: Imported items designated by blue dot (temporary) markers in the *Text Table* view.

3.7.8 Sample: Saving Translations to a Trados TM

The following is a step-by-step demonstration of how to save translations from an RC-WinTrans translation project to a Trados TM. This example shows how to save a number of items all at once. The active Trados TM must support the current project's target language (in this example the source language is English and the target language is Japanese).

① To save translations for items in the current project, first select the entry or entries in the *Text Table* view.

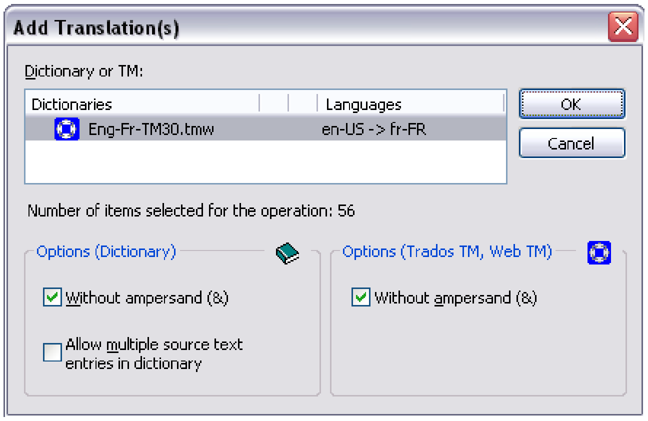
② After selecting the item(s) that you want to save, use one of the following two command:

The *Save Translation* command is available in the *Trados* toolbar and in the *Trados* menu.

The *Add Translations to Dictionary* button.

NOTE: Only translated text entries are saved. Source texts that haven't been translated will be disregarded if selected.

③ The ***Save Translation* command** saves the translation units right away to the Trados TM. The ***Add Translations* command** (in the *Dictionaries* menu) opens *the Add Translation(s)* dialog box (see figure below). Confirm the selections by clicking on the *OK* button.



Add Translation(s)

Dictionary or TM:

Dictionaries	Languages
Eng-Fr-TM30.tmw	en-US -> fr-FR

OK

Cancel

Number of items selected for the operation: 56

Options (Dictionary)

☑ Without ampersand (&)

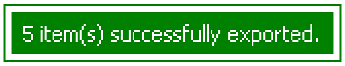
□ Allow multiple source text entries in dictionary

Options (Trados TM, Web TM)

☑ Without ampersand (&)

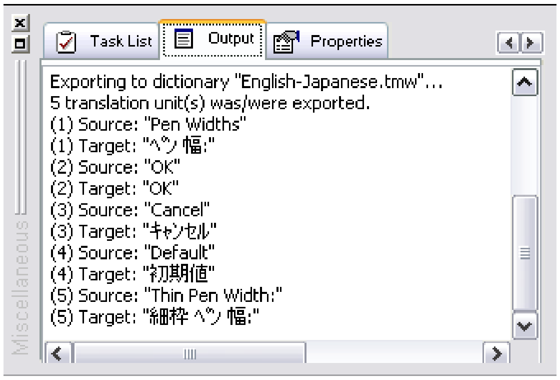
Figure 1: The *Add Translation(s)* dialog box.

Pressing the *OK* button closes the dialog box and tells RC-WinTrans to save the translations to the selected dictionary or Trados TM. The selected translation pairs will be added to the translation memory and a *Flash Info* window stating the number of exported text entries is displayed. The results of the operation will be sent to the *Output* view in the *Miscellaneous* bar.



5 item(s) successfully exported.

Figure 2: The *Flash Info* window showing the number of exported items.



```
Exporting to dictionary "English-Japanese.tmw"...
5 translation unit(s) was/were exported.
(1) Source: "Pen Widths"
(1) Target: "ﾍﾟﾝ 幅:"
(2) Source: "OK"
(2) Target: "OK"
(3) Source: "Cancel"
(3) Target: "ｷｬﾝｾﾙ"
(4) Source: "Default"
(4) Target: "初期値"
(5) Source: "Thin Pen Width:"
(5) Target: "細枠 ﾍﾟﾝ 幅:"
```

Figure 3: The *Output* tab in the *Miscellaneous* bar showing a list of items that have been exported (added) to a translation memory.

NOTE: In order to add translation to a translation memory it must first be enabled as a “save reference” (see Section 3.6.4.2 for a definition). This is the default setting for a TM that has been opened and added to the list of dictionary resources in the *Dictionaries* view.

3.8 Import and Export

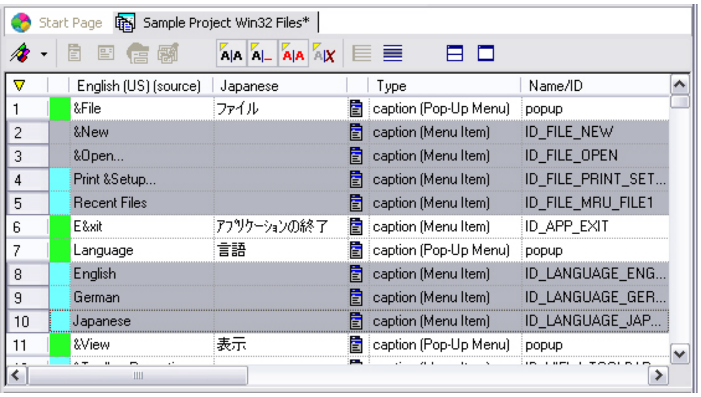
3.8.1 Overview

RC-WinTrans supports import and export operations for dictionaries and Trados™ translation memories (described in previous sections of this guide, Section 3.6.4.9 and Sections 3.7.8/3.7.9, respectively), Microsoft Excel spreadsheets (.xls), text files (.txt), translation memory exchange files (.tmx), and RC-WinTrans translation projects (.rwtproject- import only). The term “import” refers to the process of searching for matching translations and inserting them as the target texts for selected items. “Export” refers to the process of taking source texts and their corresponding translations from a translation project and adding them to one of the aforementioned file types.

RC-WinTrans’ import and export functions are based on a very simple concept: select the items for which you want to perform the import or export in the *Text Table* view, then use the appropriate menu or toolbar command. An import or export operation can be performed for just one or a number of items.

NOTE: The process of selecting items in the *Text Table* view then executing a command for those items is standard handling for other operations in RC-WinTrans as well.

Figure: RC-WinTrans’ *Text Table* view where all the non-translated items have been selected (for an operation to be performed).



▽		English (US) (source)	Japanese		Type	Name/ID
1	🟩	&File	ファイル		caption (Pop-Up Menu)	popup
2		&New			caption (Menu Item)	ID_FILE_NEW
3		&Open...			caption (Menu Item)	ID_FILE_OPEN
4	🟦	Print &Setup...			caption (Menu Item)	ID_FILE_PRINT_SET...
5	🟦	Recent Files			caption (Menu Item)	ID_FILE_MRU_FILE1
6	🟩	E&xit	ｱﾌﾟﾘｹｰｼｮﾝの終了		caption (Menu Item)	ID_APP_EXIT
7	🟩	Language	言語		caption (Pop-Up Menu)	popup
8	🟦	English			caption (Menu Item)	ID_LANGUAGE_ENG...
9	🟦	German			caption (Menu Item)	ID_LANGUAGE_GER...
10	🟦	Japanese			caption (Menu Item)	ID_LANGUAGE_JAP...
11	🟩	&View	表示		caption (Pop-Up Menu)	popup

The import and export procedure is basically the same for all the different file types. The descriptions of the operations which follow can therefore be applied to any (supported) file type, not just the ones shown in the examples. The individual file types will be discussed further on.

3.8.2 Import

Import Commands

After selecting the items for which you want to import translations in the *Text Table* view, select the appropriate *Import* command from the *File* menu, or use the *Import* button in the *Project* toolbar (Figure 1).

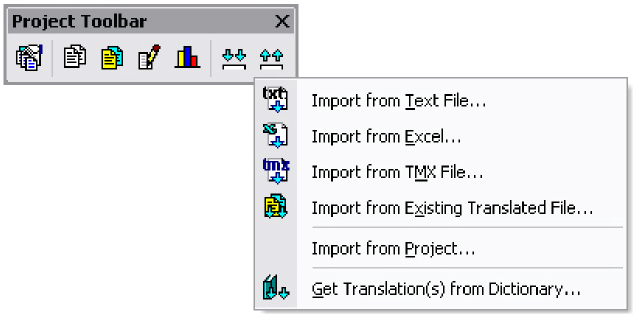
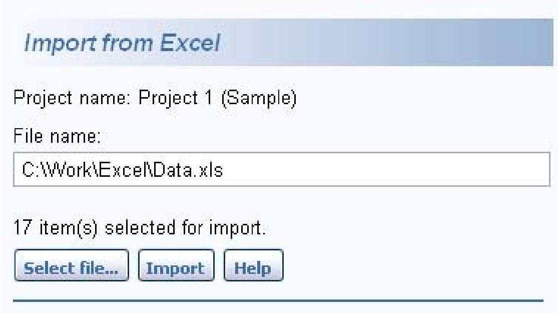


Figure 1: The *Project* toolbar with the import options displayed (with a click on the *Import* button, second from right).

Import Page

Selecting an *Import* command calls up the respective *Import* page where the file from which you want to get translations is chosen. Press the *Select file* button to browse for the file, then press the *Import* button to continue with the Import Assistant.



Import from Excel

Project name: Project 1 (Sample)

File name:

C:\Work\Excel\Data.xls

17 item(s) selected for import.

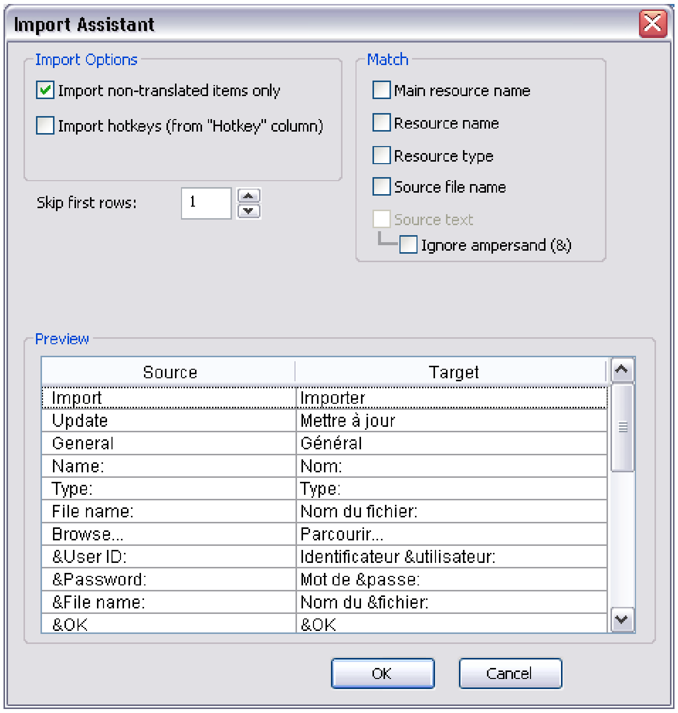
Figure 2: The *Import from Excel* page (also representative of the *Import* pages for other file types).

Import Assistant

The Import Assistant (Figure 3) is used to specify the data that is to be imported from an external file into the active translation project.

If the file from which translations are to be imported has been exported (created) with RC-WinTrans, the Import Assistant will recognize the columns and the type of data that each column contains. If the file was not created with RC-WinTrans, it may not contain a column header, in which case the Assistant will ask you to identify the data contained in each column. The columns for the source and target texts must be specified in order to import translation data.

NOTE: Unidentified (unknown) columns will display the text “<skip>” for the header in the Import Assistant’s *Preview* display. Click on the column header in the display to open a local menu where you can specify the type of data contained in the column (Figure 4).



Import Options

☑ Import non-translated items only

□ Import hotkeys (from "Hotkey" column)

Skip first rows: 1

Match

□ Main resource name

□ Resource name

□ Resource type

□ Source file name

□ Source text

□ Ignore ampersand (&)

Preview

Source	Target
Import	Importer
Update	Mettre à jour
General	Général
Name:	Nom:
Type:	Type:
File name:	Nom du fichier:
Browse...	Parcourir...
&User ID:	Identificateur &utilisateur:
&Password:	Mot de &passe:
&File name:	Nom du &fichier:
&OK	&OK

Figure 3: The Import Assistant in RC-WinTrans

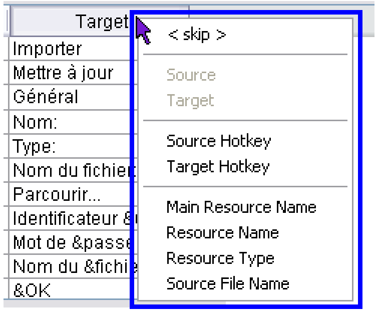


Figure 4: The local menu (called up by clicking on the column header in the Import Assistant’s *Preview* display) for specifying the column’s content.

3.8.3 Export

Export Commands

After selecting the items that you want to export in the *Text Table* view, select the appropriate *Export* command from the *File* menu, or use the *Export* button in the *Project* toolbar (Figure 1).

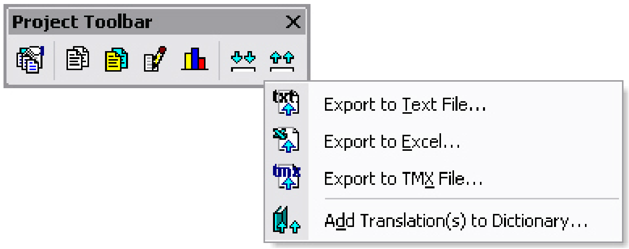
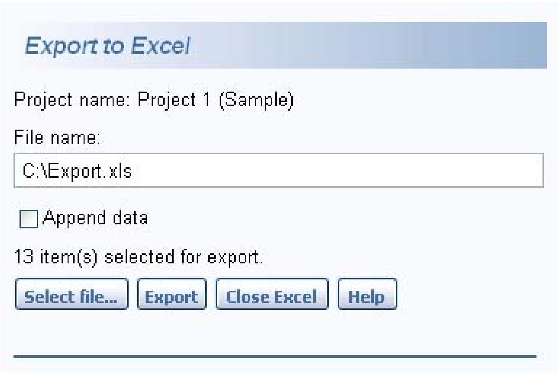


Figure 1: The *Project* toolbar with the export options displayed (by clicking on the *Export* button, far right).

Export Page

Selecting an *Export* command calls up the respective *Export* page where the file to which you want to export translations is chosen. Press the *Select file* button to browse for the file, then press the *Export* button to continue with the Export Assistant.

Figure 2: The *Export to Excel* page (also representative of the *Export* pages for other file types).



Export to Excel

Project name: Project 1 (Sample)

File name:

C:\Export.xls

□ Append data

13 item(s) selected for export.

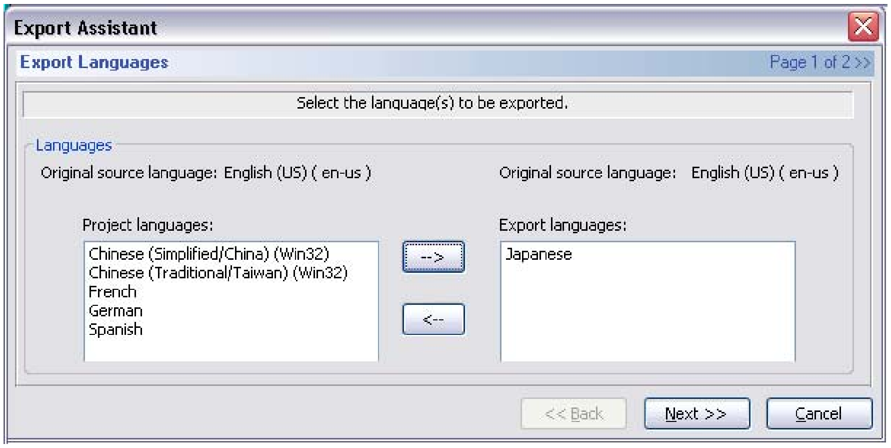
NOTE: Exported data can be appended to an existing file. It is not necessary to export all the data at once- the export can be repeated and the data appended to the file. To append data check the *Append data* option on the *Export* page.

Export Assistant

Use the Export Assistant (Figure 3) to specify the languages and the data to be exported from the active translation project to the selected file.

Languages Page

The *Languages* page is used to specify the target languages (one or more) to export. The original source language will always be exported. If at least one target language is selected the *Next* button can be used to continue on to the *Options* page.



Select the language(s) to be exported.

Original source language: English (US) (en-us)

Original source language: English (US) (en-us)

Project languages:
Chinese (Simplified/China) (Win32)
Chinese (Traditional/Taiwan) (Win32)
French
German
Spanish

Export languages:
Japanese

Figure 3: The Export Assistant's *Languages* page.

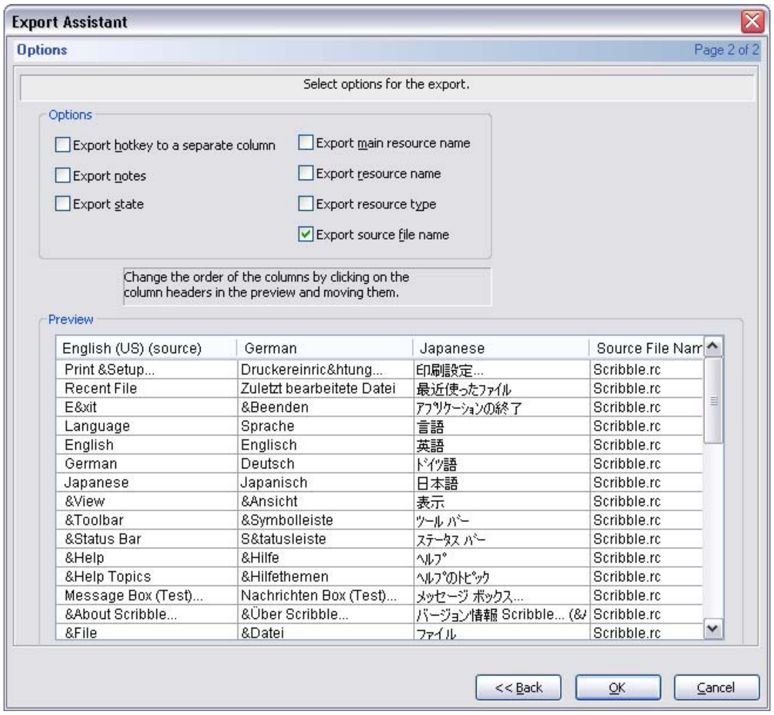
Options Page

The *Option*s page is used to specify which data will be exported and in which order (i.e., the order of the columns). In addition to the source and target texts, the following information can also be exported:

- **Hotkeys** can be removed from the the source and target texts and exported to a separate column.
- **Notes** attached to translation units.
- The **state** of a translation unit, e.g., "translated," "changed," "needs-translation," etc.
- The **main resource name/ID**, e.g., "<Dialog ID>#<Item ID>".
- The **item name/ID** (name of the resource), e.g., the ID of a dialog box control or menu item.
- The **resource type** of the translation unit.
- The **source file name** (without the path).

Columns Order

The default ordering of the columns in the destination file can be changed by clicking on a column's header in the Export Assistant’s *Preview* display and moving (dragging) the column left or right to the desired position.



Select options for the export.

Options

- ☐ Export hotkey to a separate column
- ☐ Export notes
- ☐ Export state
- ☐ Export main resource name
- ☐ Export resource name
- ☐ Export resource type
- ☑ Export source file name

Change the order of the columns by clicking on the column headers in the preview and moving them.

Preview

English (US) (source)	German	Japanese	Source File Nam
Print &Setup...	Druckereinric&htung...	印刷設定...	Scribble.rc
Recent File	Zuletzt bearbeitete Datei	最近使ったﾌｧｲﾙ	Scribble.rc
E&xit	&Beenden	ｱﾌﾟﾘｹｰｼｮﾝの終了	Scribble.rc
Language	Sprache	言語	Scribble.rc
English	Englisch	英語	Scribble.rc
German	Deutsch	ﾄﾞｲﾂ語	Scribble.rc
Japanese	Japanisch	日本語	Scribble.rc
&View	&Ansicht	表示	Scribble.rc
&Toolbar	&Symbolleiste	ﾂｰﾙ ﾊﾞｰ	Scribble.rc
&Status Bar	S&tatusleiste	ｽﾃｰﾀｽ ﾊﾞｰ	Scribble.rc
&Help	&Hilfe	ﾍﾙﾌﾟ	Scribble.rc
&Help Topics	&Hilfethemen	ﾍﾙﾌﾟのﾄﾋﾟｯｸ	Scribble.rc
Message Box (Test)...	Nachrichten Box (Test)...	ﾒｯｾｰｼﾞ ﾎﾞｯｸｽ...	Scribble.rc
&About Scribble...	&Über Scribble...	ﾊﾞｰｼﾞｮﾝ情報 Scribble... (&/	Scribble.rc
&File	&Datei	ﾌｧｲﾙ	Scribble.rc

Figure 2: The Export Assistant’s *Options* page.

3.8.4 Microsoft Excel Spreadsheets (XLS)

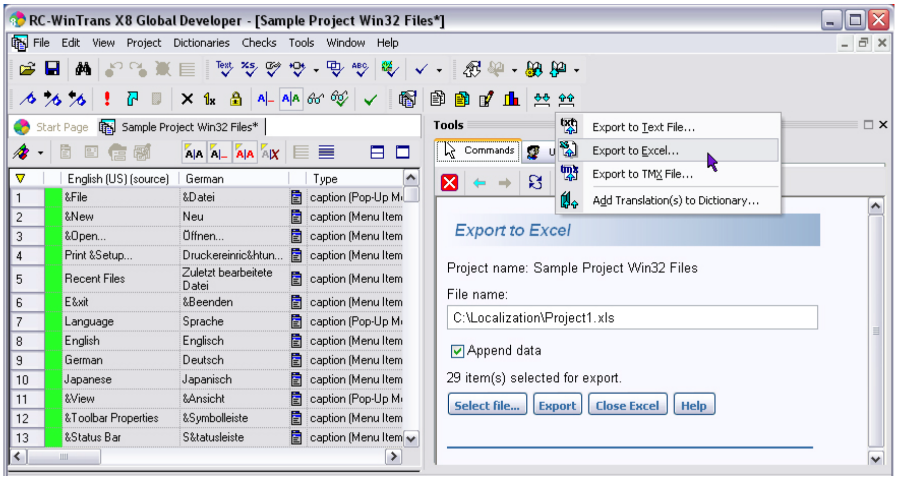
Texts can be easily sent to translators and imported (merged) back into an RC-WinTrans translation project with the use of Microsoft Excel.

RC-WinTrans’ import and export capabilities for Excel spreadsheets (file extension: .xls) provide a direct means of exchanging data and are easy to use.

There are two options for performing import and export operations from/to MS Excel: the internal Import/Export Assistants described here and the Excel Import/Export Add-On accessed via RC-WinTrans’ *Add-On* menu. The

Import/Export Add-On has its own help system with descriptions and instructions on use.

Figure 1: *Project* window with several items selected in the *Text Table* view (left) for export to an Excel spreadsheet. On the right the *Export to Excel* page has been opened as a result of selecting the *Export to Excel* command.


RC-WinTrans X8 Global Developer - [Sample Project Win32 Files*]
File Edit View Project Dictionaries Checks Tools Window Help
Start Page
Sample Project Win32 Files*
Tools
Commands
Export to Text File…
Export to Excel…
Export to TMX File…
Add Translation(s) to Dictionary…
Select file...
Export
Close Excel
Help

	English (US) (source)	German	Type
1	&File	&Datei	caption (Pop-Up M
2	&New	Neu	caption (Menu Item
3	&Open...	Öffnen...	caption (Menu Item
4	Print &Setup...	Druckereinric&htun...	caption (Menu Item
5	Recent Files	Zuletzt bearbeitete Datei	caption (Menu Item
6	E&xit	&Beenden	caption (Menu Item
7	Language	Sprache	caption (Pop-Up M
8	English	Englisch	caption (Menu Item
9	German	Deutsch	caption (Menu Item
10	Japanese	Japanisch	caption (Menu Item
11	&View	&Ansicht	caption (Pop-Up M
12	&Toolbar Properties	&Symbolleiste	caption (Menu Item
13	&Status Bar	S&tatusleiste	caption (Menu Item

Export to Excel

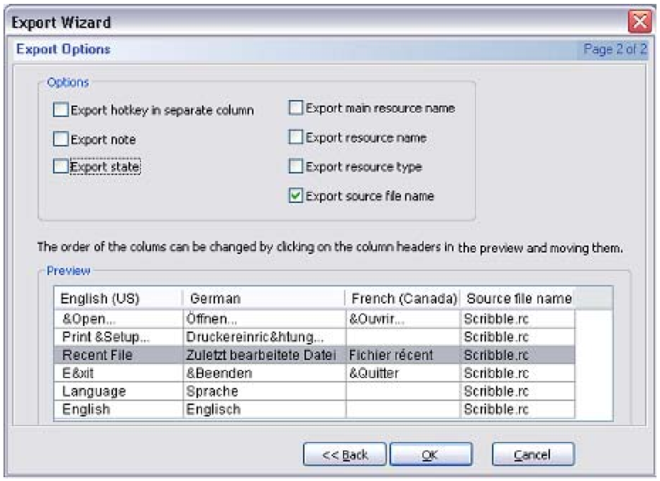
Project name: Sample Project Win32 Files

File name:

C:\Localization\Project1.xls

☑ Append data

29 item(s) selected for export.



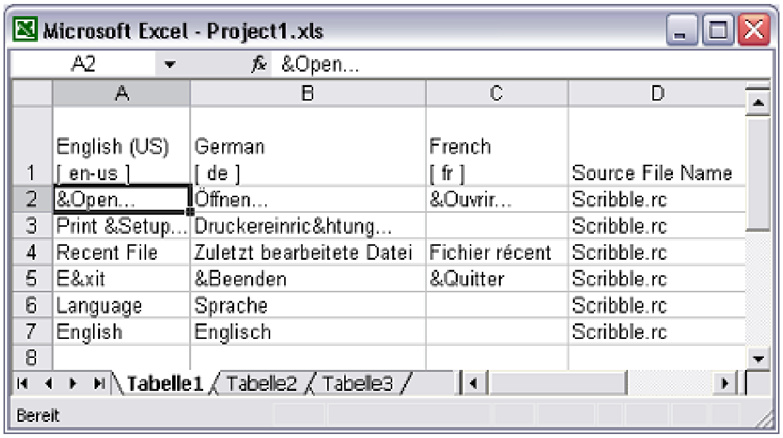
☐ Export hotkey in separate column
☐ Export note
☐ Export state
☐ Export main resource name
☐ Export resource name
☐ Export resource type
☑ Export source file name

The order of the colums can be changed by clicking on the column headers in the preview and moving them.

English (US)	German	French (Canada)	Source file name
&Open...	Öffnen...	&Ouvrir...	Scribble.rc
Print &Setup...	Druckereinric&htung...		Scribble.rc
Recent File	Zuletzt bearbeitete Datei	Fichier récent	Scribble.rc
E&xit	&Beenden	&Quitter	Scribble.rc
Language	Sprache		Scribble.rc
English	Englisch		Scribble.rc

Figure 2: The Export Assistant for exporting data to an Excel spreadsheet.

Figure 3: The Microsoft Excel application with an Excel spreadsheet containing exported text items.



	A	B	C	D
1	English (US) [en-us]	German [de]	French [fr]	Source File Name
2	&Open...	Öffnen...	&Ouvrir...	Scribble.rc
3	Print &Setup...	Druckereinric&htung...		Scribble.rc
4	Recent File	Zuletzt bearbeitete Datei	Fichier récent	Scribble.rc
5	E&xit	&Beenden	&Quitter	Scribble.rc
6	Language	Sprache		Scribble.rc
7	English	Englisch		Scribble.rc
8				

3.8.5 Text Files

Import

The *Import from Text File* command is used to import translation texts from ANSI- or Unicode- (UTF-16-LE) encoded text files.

Data Format Requirements

The form of the text data must be such that one line contains the source text (first) and the target text (second) separated by a separator character. The source and target texts **must be in quotation marks**. If the texts themselves contain quotation marks, the quotation marks need to be doubled.

Example:

There are two lines with an English source text and a German target text. A semicolon is used as a separator character.

The first line of the text file identifies each column’s language: “en-us” (English (US)) and “de” (German).

	DATA IN FILE	DESCRIPTION
1ST LINE:	“en-us”; “de”	Languages header.
2ND LINE:	“&File”; “Datei”	Source and target texts.
3RD LINE:	“File”1””; “Datei “”1””””	Source and target texts.

Export
The *Export to Text File* command is used to export text data to a text file. The encoding of the target text file can be either ANSI or Unicode (UTF-16-LE). If the file does not exist yet RC-WinTrans creates a Unicode-encoded text file by default.

The export operation writes each individual source and target text pair into one line where the source and target texts are separated by a designated separator character. Ampersand characters (& – hotkey specification) may remain in the text or be exported separately. Resource IDs and resource types are exported optionally.

When data is exported to an existing target text file it will be appended to any data already contained in the file; source and target text pairs already present in the file will be added again. The ability to append data to an existing target file lets you perform multiple export operations to collect data in a single target file.

3.8.6 Translation Memory Exchange Files (TMX)

TMX (translation memory exchange) is a file format for transporting translation text data between different tools. TMX is an XML-compliant format, so any XML- or SGML-enabled utility can be used on TMX documents. It provides a standard method of describing translation text data that is being exchanged between various tools and/or translation vendors while causing little or no loss of critical data during the process.

RC-WinTrans supports translation text data import and export operations for TMX files in the Level 1 TMX format.

See “Import from TMX File” and “Export to TMX File” topics in the RC-WinTrans Help for further information.

Chapter 4

External Translation

4 External Translation

A translation project typically have a number of target languages that are translated by one or more (external) translators. The process by which translators are provided with data to be translated and translators send translated data back for final processing is known as *external translation*.

There are three main options for providing a translator with data from an RC-WinTrans translation project:

1. The translator uses RC-WinTrans to translate the project.
2. The translator uses Microsoft Excel to translate a spreadsheet containing the text data from the project.
3. The translator is provided with a text or TMX file containing the text data from the project.

Use this chapter in conjunction with Section 3.8, “Import and Export,” to determine the best method of supplying a translator with data to be localized.

4.1 Translation with RC-WinTrans

If a translator is working with RC-WinTrans a translation project can be exported as a standard ZIP file containing all the necessary information including the source data to be translated and other supplemental data. The translator opens the ZIP file with RC-WinTrans (extracting the translation project), translates the project, then sends it back (also as a ZIP file) to be merged into the main translation project.

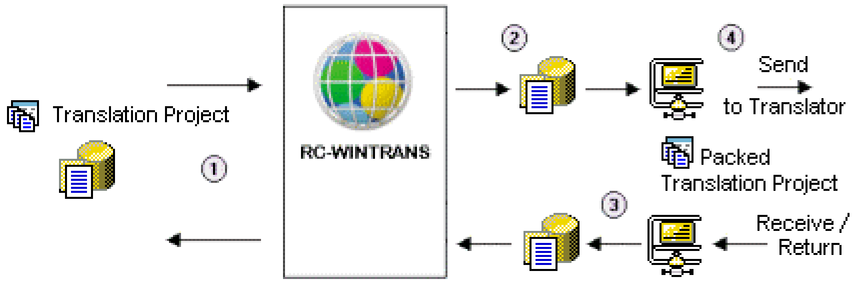
In lieu of packing data into a ZIP file it is also possible to exchange the .fsmdb project file directly.

4.1.1 Exporting a Translation Project (for a Translator)

If the translator is in posession of RC-WinTrans a translation project can be created and sent out for translation, then returned upon completion.

Extract a project for a translator from the main translation project using the *Export Translation Project for Translator* command from the *File* or *Project* menus.

Figure: Diagram demonstrating the process of extracting a translation project (from the main translation project) to send to a translator and reintegrating that project after it has been translated and returned.



Creating a Translation Project for a Translator

① **Translation project.**
A (the main) translation project, typically with several files to be translated and several target languages.

② **Translation project created/exported for translator.**
All relevant data is exported to create a translation project for a translator. This is done with the Export Translation Project Assistant, accessed via the *Export Translation Project for Translator* command in the *File* (or *Project*) menu.

The translation project for the translator (2) is a copy of the main project (1) and usually contains only those target languages that will be edited by a certain translator.

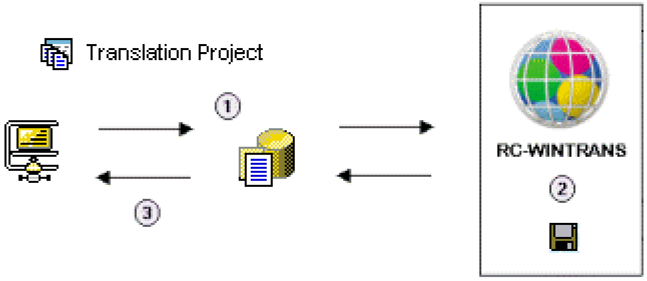
The data contained in the translation project for the translator is specified with the Export Translation Project Assistant. The file extension for the target file (exported database) is **.fsmdb**.

③ **Project packed and ready.**
The translation project is packed into a (standard) ZIP file to be sent to a translator.

④ **Translation project translated and returned, merged with other data.**
The translator opens the ZIP file to extract the translation project, which s/he then translates in RC-WinTrans and sends back to be merged into the main project.

4.1.2 Receiving and Returning a Translation Project

Figure: A packed translation project is received then opened and translated with RC-WinTrans. After the project is translated it is again packed into a ZIP file to be returned to the project manager or home office.



① **Receive and open (packed) translation project (extract from ZIP file).**
Open (unpack) the translation project by selecting the *Open* | *Packed Translation Project (ZIP File)* commands from the *File* menu.

② **Project review and update by translator.**
The unpacked translation project is translated with RC-WinTrans.

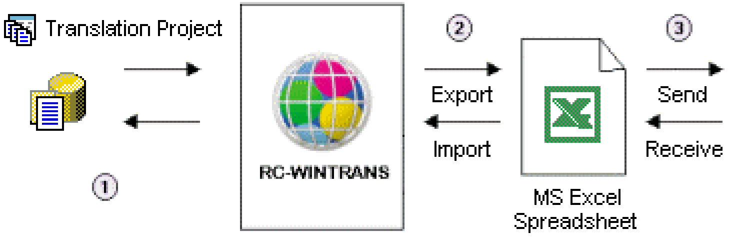
NOTE: After unpacking (unzipping) the translation project RC-WinTrans gives you the option of opening and automatically adding it to the *Projects Overview* tab in the *Workspace* bar. If the project is not added to the *Projects Overview* tab right away, use the *Open* | *Project* commands from the *File* menu to open the database file manually (file extension: .fsmdb).

③ **Returning a Translated Project**
In the *File* or *Project* menu select the command *Export Translation Project for Translator*. The Export Translation Project Assistant opens to guide you through the process of creating a packed translation project (ZIP file format) to return to the distributor.

4.2 Translation with Microsoft Excel

Microsoft Excel can be used to translate an Excel spreadsheet that has been created with RC-WinTrans. An Excel spreadsheet containing the text to be translated can be sent to a translator and returned with the translations to be imported back into the translation project.

Figure: Text to be translated is exported to a Microsoft Excel spreadsheet. The ranslation texts from the spreadsheet are then imported back into the main project.

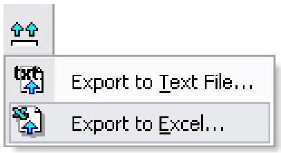


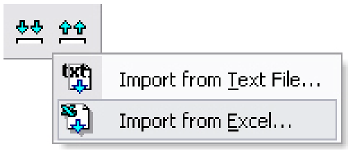
① Translation Project

A translation project with several files and several languages. The data of translation units can be exported to an Excel spreadsheet and translation text can be imported into the RC-WinTrans project.

② Export to/Import from Excel Spreadsheet

Select the items to export in the *Project* window's *Text Table* view and use the *Export to Excel* command (*Export* toolbar button, shown right, or located in the *File* and *Project* menus) to export them to an Excel spreadsheet.





Use the *Import from Excel* command (*Export* toolbar button, shown left, or located in the *File* and *Project* menus) to import translated text into the project.

③ Send/Receive Data (Spreadsheet)

The Excel spreadsheet is sent to the translator to be translated in Microsoft Excel. When the translated data is returned it is imported back into the the translation project.

Please refer to Section 3.8.4 of the guide for more information on importing and exporting data to/from Microsoft Excel spreadsheets.

NOTE: The import and export processes can also be carried out by the MS Excel Import/Export Add-On accessed via RC-WinTrans' *Add-On* menu. The Import/Export Add-On has its own help system with descriptions and instructions on use.

4.3 Translation with Text or TMX Files

A text file or TMX file can be exported from RC-WinTrans to provide a translator with translation data. The translator can use a translation memory system (e.g., SDL TRADOS™ or STAR Transit) to translate the file. When the translated text or TMX file is received, the data can be imported back into the RC-WinTrans translation project.

Chapter 5

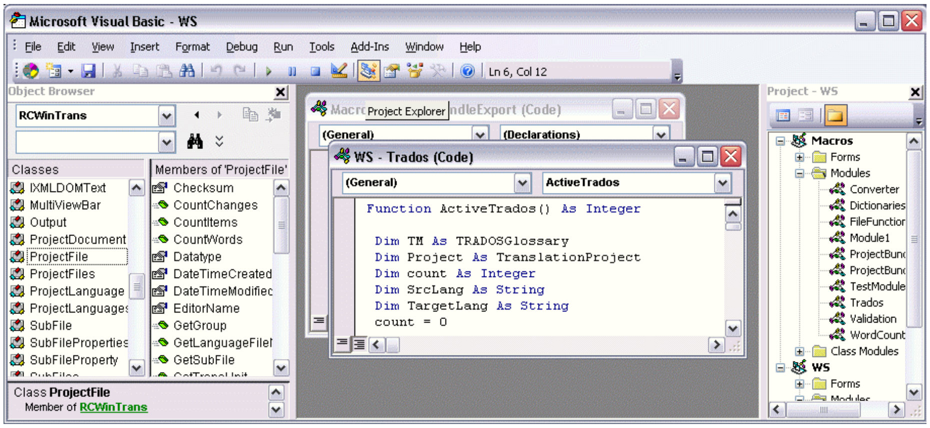
RC-WinTrans System Design

5 RC-WinTrans System

5.1 Microsoft Visual Basic for Applications

The RC-WinTrans localization environment comes integrated with Microsoft Visual Basic for Applications (VBA). Together with the RC-WinTrans COM (Component Object Model), VBA technology helps make RC-WinTrans customizable, extendable, open for new solutions and capable of meeting changing requirements in software localization and workflow processes.

Microsoft Visual Basic for Applications is a complete development environment, with editor and programming code debugger for developing and oping and running VBA macros. Thus macros can be used to combine the powerful RC-WinTrans object model with a wide range of existing COM components to create new solutions. To open the VBA integrated development environment (IDE) use the *Visual Basic Editor* command from the *Tools* menu, or click on the *Visual Basic Editor* button (shown above) in the *VBA* toolbar.



```
Function ActiveTrados() As Integer

 Dim TM As TRADOSGlossary
 Dim Project As TranslationProject
 Dim count As Integer
 Dim SrcLang As String
 Dim TargetLang As String
 count = 0
```

Figure: The Microsoft VBA development environment.

5.1.1 VBA IDE Languages – GUI and VBA Help

The graphical user interface (GUI) and the Help system of the integrated Visual Basic for Applications is available in 12 languages: Chinese (Simplified), Chinese (Traditional), Dutch, English, French, German, Italian, Japanese, Korean, Portuguese, Spanish, and Swedish. The language to be used is specified on the *User Interface* property page in the *Options* dialog box. The *Options* dialog box is opened with the *Options* command in RC-WinTrans' *Tools* menu.

5.1.2 The RC-WinTrans "Application" Object

The RC-WinTrans automation objects (COM (Component Object Model)) can be used in the integrated VBA environment. RC-WinTrans' main (or "root") object is the "Application" or "RC-WinTrans" object (both names can be used). Beginning with the root object, RC-WinTrans' object model branches out to encompass all the different parts of the RC-WinTrans application, right down to the individual objects and methods.

Example (object): "**RCWinTrans**.ActiveProject.Name"

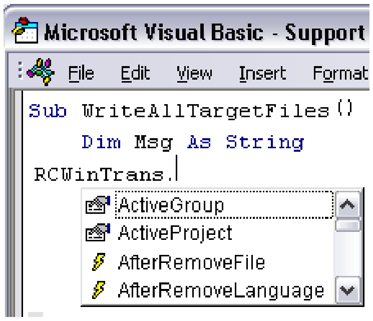
The RC-WinTrans COM object model is discussed in Section 5.3 of this guide.

5.1.3 Help for VBA Objects

Direct help is available from within the VBA IDE (**I**ntegrated **D**evelopment **E**nvironment) for all RC-WinTrans objects, methods, parameters, member variables, enumerations, and events. It is accessed using one of the following methods:

F1 Function Key Use the F1 function key to open the help for the selected object or method. To do this, select (or give the focus to) the object or method (the text in the Editor) and press the F1 key.

IntelliSense RC-WinTrans objects are enabled for use with the *IntelliSense* mechanism (see figure). IntelliSense allows you to get help on elements without leaving the code editor. You can remain in the current context, find the information you need, and insert language elements directly into your code. You can even have IntelliSense complete your typing for you.



```
Sub WriteAllTargetFiles()
    Dim Msg As String
RCWinTrans.
```

Figure: The VBA IDE in which IntelliSense has automatically opened a list of available methods and events for the current RC-WinTrans object.

5.2 VBA Macros

System Macros
RC-WinTrans initially comes with a number of VBA macros files which are called RC-WinTrans system macros files. Each of these files contains a number of macros modules with uses that make up a substantial part of RC-WinTrans' functionality.

RC-WinTrans system macros:

VBA PROJECT (MACROS FILE)	DESCRIPTION / VBA IMPLEMENTATION
"SystemMacros"	The main macro of RC-WinTrans' system functionality and helper functions.
"ExcelMacros"	Import and export functions for Microsoft Excel.
"ResxEditorMacro"	Functionality for using the External .NET Forms Editor.
"Dictionary"	Functionality for dictionaries.
"FileConverters"	Several file converters.
"XMLConverter"	XML files converter.

System macros files are maintained by Schaudin.com. Changing the code in the system macros is not recommended since modifications get lost the next time that the system macros file is updated by a newer version supplied by Schaudin.com.

NOTE: Support for the functionality of system macros is only guaranteed if the code in this macros file is unchanged.

Additional macros files (user macros) can be added to RC-WinTrans' VBA system. Macros files are placed (by default) in the "\VBAMacros" directory of the RC-WinTrans installation directory. See "User Macros Files" further on.

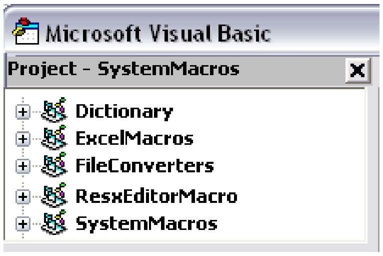


Figure 1: The Project Explorer window of the VBA development environment showing the RC-WinTrans system macros.

Excel Macros

The Excel macros behave like the system macros with the specification that they are used exclusively with Microsoft Excel (97, 2000, and XP). Macros with functionality that uses Excel are placed in this separate macros file for maintenance purposes.

User Macros Files

User macros can be created from scratch or existing macros can be added to RC-WinTrans' VBA system. There is no limit to the number of user macros files that can be added to the system.

To create or remove a user macros file, use the *Options* command from the *Tools* menu in RC-WinTrans. When the *Options* dialog box opens click on the *VBA Macros Files* property page.

NOTE: User macros files cannot be created, added, or removed with the VBA development environment. These tasks are performed using RC-WinTrans' *VBA Macros Files* property page.

Macros Directory

The default directory for all macros files is the "VBAMacros" subdirectory in the RC-WinTrans installation directory. This is where the system macros files are initially placed. Using this directory for all user macros files is recommended but not mandatory- other directories may be used as well.

The "VBAMacros" directory also contains an INI file named "RCWinTrans_MacroFiles.ini." This is an initialization file containing a list of all the user macros files to be loaded into RC-WinTrans' VBA system when RC-WinTrans starts up.

Executing Macros

A macro can be executed by performing the following steps:

1. Open the *Run Macro* dialog box (Figure 3) by clicking on the *Run Macro* toolbar button (shown right) or by selecting the *Macros* command in the *Tools* menu. ▶
2. In the *Run Macro* dialog box, specify the name of and select the macros file.
3. Press the *Run* command button at the bottom of the dialog box.

Once a macro has been executed via the dialog box, it can be run again directly using the *Run Macro* button (Figure 2) in the *VBA* toolbar, or by selecting the *Macros* command from the *Tools* menu.

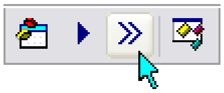
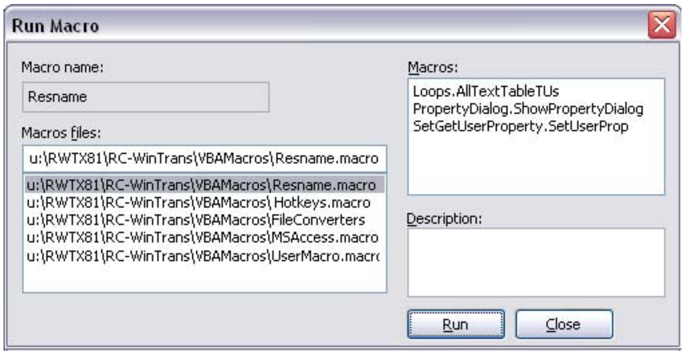


Figure 2: The *Run Macro* button selected in the *VBA* toolbar.



Run Macro

Macro name: Resname

Macros files: u:\RWTX81\RC-WinTrans\VBAMacros\Resname.macro

- u:\RWTX81\RC-WinTrans\VBAMacros\Resname.macro
- u:\RWTX81\RC-WinTrans\VBAMacros\Hotkeys.macro
- u:\RWTX81\RC-WinTrans\VBAMacros\FileConverters
- u:\RWTX81\RC-WinTrans\VBAMacros\MSAccess.macro
- u:\RWTX81\RC-WinTrans\VBAMacros\UserMacro.macr

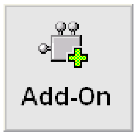
Macros:

- Loops.AllTextTableTUs
- PropertyDialog.ShowPropertyDialog
- SetGetUserProperty.SetUserProp

Description:

Figure 3: The *Run Macro* dialog box.

5.3 Add-Ons



The RC-WinTrans localization environment comes integrated with Microsoft Visual Basic for Applications (VBA). Together with the RC-WinTrans Component Object Model, VBA technology helps make RC-WinTrans customizable, extendable, open for new solutions and capable of meeting changing requirements in software localization and workflow processes.

Add-ons are program utilities that enhance a primary program. In RC-WinTrans, it is a software component that extends the application's functionality.

Technically, an RC-WinTrans add-on is a VBA macros file (file extension: .macro) with at least one procedure implemented for execution. The functionality is coded using Microsoft VBA and can do virtually anything a user (third-party manufacturer or Schaudin.com) needs it to do in order to extend RC-WinTrans' capabilities to suit their particular requirements.

RC-WinTrans add-ons extend RC-WinTrans' existing capabilities to carry out specialized tasks and functions not already included in the application. Add-ons have their own menu in RC-WinTrans to access the specified functionality (see "*Add-On* menu," below). The integration is seamless and can be realized with a minimum of time, cost and effort thanks to RC-WinTrans' advanced technological design.

Activating an Add-On in RC-WinTrans
To activate an add-on, simply copy it (the VBA macros file, file extension: **.macro**) to the "Add-Ons" folder in the RC-WinTrans installation directory:

<Install Dir>\RC-WinTrans X8\Add-Ons\

Once the add-on is available in/accessible from the "Add-Ons" folder, RC-WinTrans will automatically load the add-on into its VBA system at start-up.

Accessing Add-On Functionality: The *Add-On* Menu
The special functionality of a given add-on is accessed via RC-WinTrans' *Add-On* menu (see figure). An add-on component usually adds one or more menu items (commands) to the *Add-On* menu.

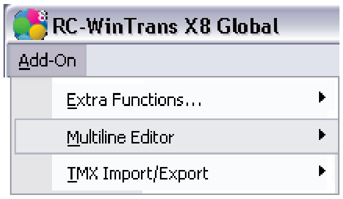


Figure: The *Add-On* menu in RC-WinTrans.

Add-Ons Available for RC-WinTrans

There are a number of add-ons that have already been created for RC-WinTrans. Some can be downloaded for free from the customer download area of our Website while others must be purchased before they can be accessed. The following list represents a selection of these:

- ***Microsoft Excel Import/Export***
  An add-on rendering the process of importing and exporting data from and to an MS Excel spreadsheet more direct, faster and easier. It also lets you determine whether the import/export operations will be performed for currently selected items, for the active project file, or for the entire translation project (all files at once). (Available via the *Add-On* menu.)

- ***Google Machine Translation***
  Accesses machine translations using the Google Translation Web interface to quickly supply translation texts while working with RC-WinTrans. Further information about this add-on can be found on our Website.

- ***Multiline Editor***
  Extends RC-WinTrans' functionality to display and edit source and translation texts containing new line and tab control characters.

- ***Extra Functions***
  This add-on extends the RC-WinTrans application with a set of functions created to accommodate certain user requests and suggestions such as a maximum string length check while editing. (Available via the *Add-On* menu.)

- ***TMX Import/Export***
  Allows data to be exported as a TMX format file and text data from a TMX file to be imported into RC-WinTrans.

Obtaining a Custom Add-On

Contact Schaudin.com for information on and assistance with creating a custom RC-WinTrans add-on to implement specialized functionality.

5.4 RC-WinTrans COM Object Model

RC-WinTrans has a COM (**c**omponent **o**bject **m**odel) automation interface which allows access to translation projects, the files that a project contains, and all translation data down to the individual attributes. The object model also provides access to RC-WinTrans' GUI (**g**raphical **u**ser **i**nterface) components and "Tools" objects.

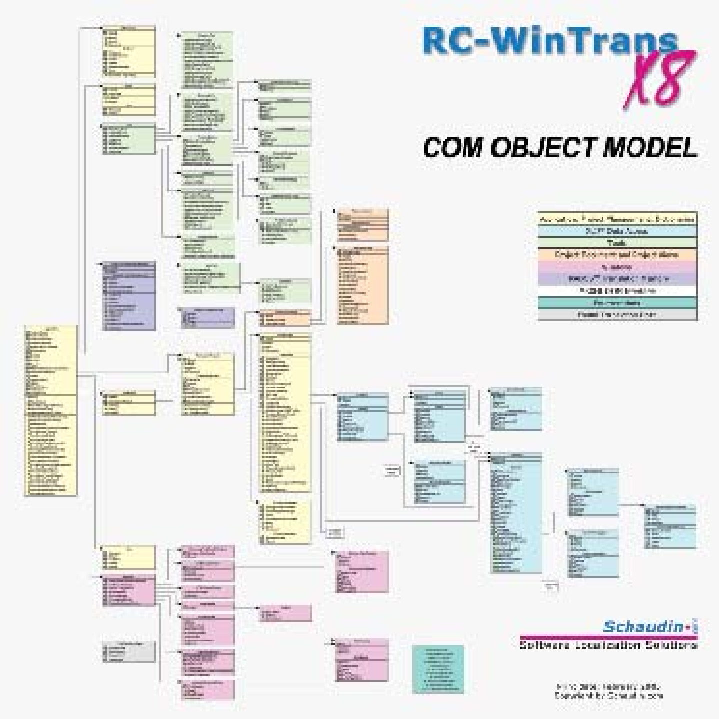


Figure: The RC-WinTrans component object model (printed poster included with the Global Edition and available upon request).

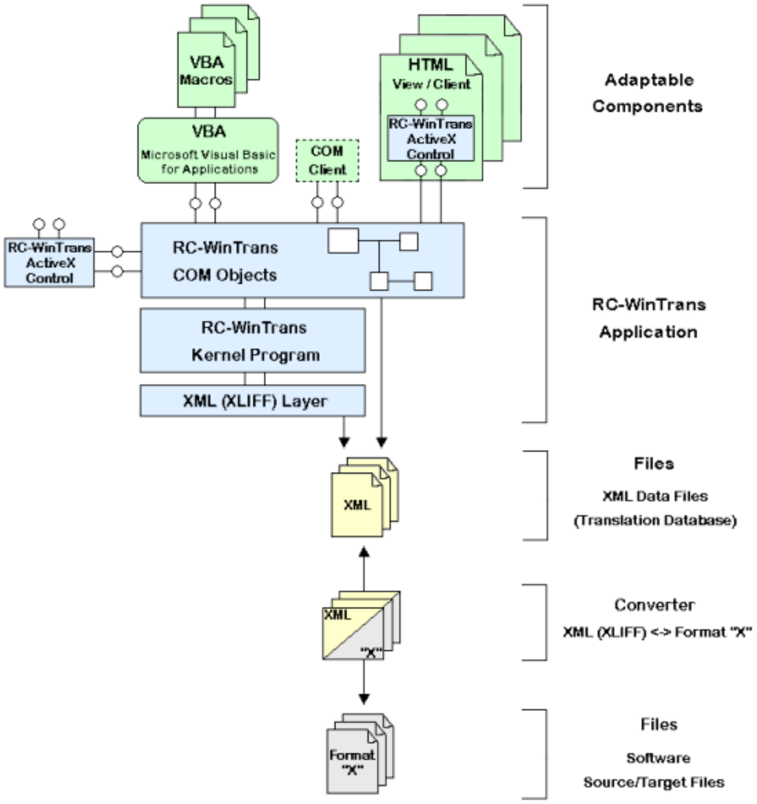
RC-WinTrans can be controlled/directed by other applications via OLE automation. Thus RC-WinTrans can be fully integrated with detailed constructs such as a development environment or a workflow management system. Individual tasks that had to be performed manually in the past can now be automated and new functionalities can be realized.

All major RC-WinTrans application GUI components, projects, and translation data can be accessed by other applications and from within the integrated VBA (Visual Basic for Applications) engine via the RC-WinTrans COM.

5.5 System Architecture

Two Data Storage Types: Database or XML (XLIFF) Files
Post-Version 7, RC-WinTrans was designed to handle software localization based on a single file format: the XLIFF file. This format is an open, **XML-compliant format** (XLIFF = **X**ML **L**ocalization **I**nterchange **F**ile **F**ormat) that is non-proprietary and well-suited to localization purposes. The general concept is that RC-WinTrans processes files to be localized/translated as XLIFF data. RC-WinTrans uses XLIFF files as translation data files (database) where the translation data of the respective source files is stored.

RC-WinTrans X8 System Architecture Model



In November 2007 the RC-WinTrans system was expanded to accommodate a **(Microsoft Access) database format** to store project data. The database storage option is offered in addition to the standard one based on XLIFF files. This method is convenient because it keeps all the project data together in one single file (file extension: .fsmdb), whereas an XML-based project will contain multiple XLIFF files, increasing with the number of source files and languages contained in the project.

RC-WinTrans Application

The RC-WinTrans application itself can be divided into three parts. First there is the RC-WinTrans kernel program with the essential application programming frame including the GUI (**g**raphical **u**ser **i**nterface). Then there are the COM objects that form the automation interface of RC-WinTrans and that make objects available for accessing the XLIFF translation data right down to individual attributes- all data is accessible including data added by the user. These objects allow for project handling and for GUI access to several tools and more. They can be used through the integrated VBA (Visual Basic for Applications) and by automation clients using RC-WinTrans as the automation server. The third part of the RC-WinTrans application is the XLIFF layer offering general access to the XLIFF translation data files.

The RC-WinTrans ActiveX control is an extension of the RC-WinTrans application. It represents the technical component of RC-WinTrans for the user of the ActiveX control.

VBA and Adaptable Components

Microsoft VBA is an essential part of RC-WinTrans. It has been integrated into the application and is included as a part of it so that functionality can easily be created, supplemented, or modified.

RC-WinTrans comes equipped with several VBA macros. These are primarily macros with specific functionalities that also have an open code so that they may be customized if desired. Examples of these include macros for word count and check functions, support for additional file types, and command line support that can be extended with custom commands for maximum flexibility.

Custom VBA macros can be created with the VBA Editor provided and the event handler can be used to catch events fired by RC-WinTrans so that these macros can be included in RC-WinTrans' activities and processes. See Section 5.2, “VBA Macros,” and Section 5.3, “Add-Ons” for further information.

COM Client

An automation client can use RC-WinTrans as an automation server. The RC-WinTrans COM (**c**omponent **o**bject **m**odel) objects (automation interface) provide full access to the data and to RC-WinTrans’ GUI including the use (calling, running) of VBA macros.

Data Storage Files (XML Data Files/Microsoft Access Database)
The XML (XLIFF) data files holding the translation data are created with converters that convert a file in format “X” (unknown to RC-WinTrans) into the XLIFF format that is processed by RC-WinTrans. The files are managed by RC-WinTrans through projects where a single project typically has a number of source files with their respective XLIFF data files (file extension: **.rwtxlf**).

RC-WinTrans can also store translation data in a (Microsoft Access) database (file extension: **.fsmdb**).

File Converters
File converters are components, external programs or VBA macros that convert a source file in format “X” into an XLIFF file so that it can be processed by RC-WinTrans. Any type of file can theoeretically be converted to the XLIFF file format with the use of converters (built-in or custom-made) and thereby displayed and edited in RC-WinTrans.

Adaptable file converters allow RC-WinTrans to be used to translate a wide range of files insofar as such a converter is available or can be made. The task of a file converter is to read a file in format “X” and to write an XLIFF file. Vice versa a file converter writes a target file in format “X” using the translated/localized data of an XLIFF file.

For additional details see the “File Converters” topic in the RC-WinTrans Help.

Files (Software: Source and Target Files)
These are the files to be translated/localized. RC-WinTrans does not process source files directly. The task of reading and writing the source and target files is done by file converters specialized in reading and writing a certain file format.

HTML Views and Clients
HTML pages can be used to create custom views and functionality. The RC-WinTrans application can be “integrated” into an HTML page via the RC-WinTrans ActiveX control.

Chapter 6

Command Line Processing

6 Command Line Processing

RC-WinTrans offers two choices to automate processes via batch processing: the command line tool and Windows scripting.

Command Line Tool

RC-WinTrans has a command line tool ("**RWTCmd.exe**") which offers a set of command line commands to be called from the command prompt or (usually) from a batch process. The command line tool is a small program which itself does not execute a command. Instead, the command line tool uses RC-WinTrans as an automation server and the command is executed by RC-WinTrans.

A list of the available command line commands can be found in the RC-WinTrans Help topic "Command Line Commands."

Windows Scripting

RC-WinTrans supports the use of script files in any language supported by the Windows Scripting Host. Examples of such a scripting language include VBScript, Jscript, and PerlScript. A script can make use of RC-WinTrans' extended COM automation interface to offer a wide variety of options for automating processes.

For further details see the topic "Scripting Reference" in the RC-WinTrans Help.

Extendibility

Both command line interface types "batch" and "script" can be extended with Microsoft Visual Basic for Applications (VBA) which is integrated into the RC-WinTrans localization environment. This means that the set of available commands can be extended with VBA procedures.

Samples

RC-WinTrans comes with samples for using the command line tool in batch files and for using Windows Scripting. The samples can be found in the "Samples" folder in the RC-WinTrans directory ("Install Dir>\RC-WinTrans X8\samples\commandline\"). Consult the RC-WinTrans Help for a list of samples.

Chapter 7

Contact and Support

7 Contact and Support

We at Schaudin.com strive to make our products and documentation comprehensive and easy to use. Should you have additional questions, here are some other ways to get help:

Online (www.schaudin.com)
Find additional product information, help files, release notes and more on our Website at www.schaudin.com. You can also access free downloads including the latest RC-WinTrans updates using your product license key in the site's customer download area.

E-mail: Direct your sales, support, and general information inquiries to us via e-mail using one of the addresses provided below. Please supply all relevant information and questions so that we can direct your message accordingly.

Phone: In addition to free online support, we also provide complimentary tele-phone support for our customers in English and German. Our phone support hours are 10 a.m. – 5 p.m. middle European time (GMT+1) at the following number: +49 6071 951706. Both our European and North American locations can be contacted with sales and general information inquiries during their respective regular business hours.

Schaudin.com (Germany)		**Schaudin.com North America**	
(Support, Sales & General Information)		***(Sales & General Information)***	
Address:	Ritterseestr. 29 64846 Gross-Zimmern Germany	**Address:**	6900 California Ave. SW #503 Seattle, WA 98136-1984 USA
Phone:	+49 6071 951706	**Phone:**	1-206-935-5070
Fax:	+49 6071 951707	**Fax:**	1-206-935-5075
E-mail:		**E-mail:**	
Info:	info@schaudin.com	***Info:***	info@schaudin.com
Sales:	sales@schaudin.com	***Sales:***	sales@schaudin.com
Support:	support@schaudin.com		