

Integrated Translation Environment overview

[See also](#)

The Integrated Translation Environment (ITE) is a suite of tools that facilitates software localization and simultaneous development of applications for different locales. Because of its integration with the Delphi IDE, the ITE lets you manage multiple localized versions of an application as part of a single Delphi project group.

The ITE creates and maintains a separate resource DLL for each localized version of the application under development. It provides a grid for viewing and editing translated resources. After modifying your translations, you can update all of your project's resource DLLs with a single menu command. You can also create a central repository for translations that you want to reuse in different projects.

The ITE consists of three tools:

- The [Translation Manager](#) displays a grid for viewing and editing translated strings and other resources.
- The [Translation Repository](#) provides a central database for translations that can be shared across projects and by different developers.
- The [Resource DLL wizard](#) generates and manages resource DLLs for the languages you select.

You can run localized versions of your application in debug mode by setting the [active language](#).

The Resource DLL wizard organizes the localized DLLs associated with an application into a *project group*. When you use the ITE, it is strongly recommended that you work through the [Project Manager](#), always opening the project group first and then selecting the specific forms and units you want to modify.

To configure ITE options, choose Tools|Environment Options|Translation Tools.

Topics

[About the Resource DLL wizard](#)

[Creating and updating resource DLLs](#)

[Launching the Resource DLL wizard from the File menu](#)

[Adding languages to a project](#)

[Updating resource DLLs](#)

[Removing languages from a project](#)

[Restoring languages to a project](#)

[Setting the active language](#)

[Using the Translation Manager](#)

[Editing in the Translation Manager](#)

[Using the Multiline editor](#)

[Keyboard shortcuts](#)

[Using the Repository](#)

[Using the Auto Translate option](#)

[Setting up a translation repository](#)

[ITE-generated files](#)

About the Resource DLL wizard

[See also](#)

The Resource DLL wizard presents different menus and options depending on how it is launched. Use the Resource DLL wizard to

- [create and update resource DLLs.](#)
- [add resource DLLs to a project.](#)
- [remove resource DLLs from a project.](#)
- [set a project's active language.](#)

To configure options for the Resource DLL wizard, choose Tools|Environment Options|Translation Tools.

Creating and updating resource DLLs

[See also](#)

Before you can use the Translation Manager or Repository, you must add languages to your project by running the Resource DLL wizard.

[Launching the Resource DLL wizard from the File menu](#)

[Adding languages to a project](#)

[Updating resource DLLs](#)

[Removing languages from a project](#)

[Restoring languages to a project](#)

Launching the Resource DLL wizard from the File menu

[See also](#)

Launching the Resource DLL wizard from the File menu lets you add languages to your project and remove languages from your project in a single step. It also gives you an opportunity to include extra files, such as .RC files, that the IDE doesn't know about. If you have multiple projects open in the IDE, you can process several at once.

To use the Resource DLL wizard,

1. Save and build your project.
2. With your project open in the IDE, choose File|New, double-click on Resource DLL Wizard, then click Next.
3. Select your project from the list that appears in the dialog. (That is, make sure a check mark appears in the box to the left of each project you want to process.) At this point, you can also set the root directory for the DLL projects and change the base-language setting (for Repository queries). Then click Next.
4. Select the languages you want to include in your project by checking the boxes next to them. To remove a language, uncheck its box.
Select only languages which are translation targets. You don't need to create a resource DLL for the primary (base) language in which you are developing your application, but you can do so if you wish. You can change the DLL extension for any language by clicking in the Extension column.
Click Next.
5. Set the directory path for each language's resource DLL by clicking and typing in the Path column. The directories you specify must fall under the root directory selected in Step 3 above.
Click Next.
6. To include additional files in your main (application) project, click Add File. When you are done adding files, click Next.
7. The languages you've selected appear in the next dialog. For each language,
 - If a resource DLL already exists for the language (in the directory you have specified), click in the Update Mode column to select Update or Overwrite. Choose Update to keep and modify the existing DLL project; choose Overwrite to create a new, empty DLL project. **If you choose Overwrite, the wizard will delete the old DLL project and any translations it contains.**
 - If no resource DLL exists, "Create New" appears in the Update Mode column. There are no further options to select.
Click Next.
8. Click Finish to create or update resource DLLs for the languages you've selected and include them in your project, and to remove from your project any languages you've unselected. (When you remove languages, the wizard removes their resource DLLs from your project, but it does not delete the DLLs or the directories they reside in.)

- If the wizard asks to generate a DRC file, click Yes. Any project that uses its own resource strings (instead of previously compiled RC files) needs a DRC file.

- If you are sure that no new DRC files are needed (because your project doesn't introduce any resource strings of its own), you can select the "Skip DRC files that are not found" check box in the final dialog. This prevents the wizard from generating, or asking to generate, DRC files.

Later, you can add languages to and remove languages from your project by rerunning the Resource DLL wizard.

To [restore a language](#) after removing it, choose Project|Languages|Add and run the wizard again.

The Resource DLL wizard creates a separate DLL project for each language you have selected. If Automatic Repository Query is checked on the Translation Tools page of the Environment Options dialog (Tools|Environment Options), the wizard populates resource DLLs with translations for any strings that have matches in the Repository. Otherwise, the generated projects contain untranslated copies of the resource strings in your original project. Your next step is to translate these strings in the [Translation](#)

Manager and build your DLLs.

Because all the generated projects have the same name, they must be kept in separate directories. Do not attempt to edit the generated project (.DPR) files, since your changes will be overwritten the next time the DLL is updated.

References to the generated projects are maintained in the parent project's options (.DOF) file. The Resource DLL wizard also creates a project group that includes the parent project and each of the generated DLL projects. It's a good idea to save and use this project group, since it simplifies the task of recompiling the resource DLLs.

Adding languages to a project

[See also](#)

You can add new resource DLLs to a project at any time. If you have multiple projects open in the IDE, you can process several at once.

1. Save and build your project.
2. With your project open in the IDE, choose Project|Languages|Add. When you open the Resource DLL wizard this way, it displays only languages that are not yet included in your project.
3. Select your project from the list that appears in the dialog. (That is, make sure a check mark appears in the box to the left of each project you want to process.) At this point, you can also set the root directory for the DLL projects and change the base-language setting (for Repository queries). Then click Next.
4. Select the languages you want to include in your project by checking the boxes next to them.
Select only languages which are translation targets. You don't need to create a resource DLL for the primary (base) language in which you are developing your application, but you can do so if you wish. You can change the DLL extension for any language by clicking in the Extension column.
Click Next.
5. Set the directory path for each language's resource DLL by clicking and typing in the Path column. The directories you specify must fall under the root directory selected in Step 3 above.
Click Next.
6. The languages you've selected appear in the next dialog. For each language,
 - If a resource DLL already exists for the language (in the directory you have specified), click in the Update Mode column to select Update or Overwrite. Choose Update to keep and modify the existing DLL project; choose Overwrite to create a new, empty DLL project. **If you choose Overwrite, the wizard will delete the old DLL project and any translations it contains.**
 - If no resource DLL exists, "Create New" appears in the Update Mode column. There are no further options to select.
Click Next.
7. Click Finish to create or update resource DLLs for the languages you've selected and include them in your project.
 - If the wizard asks to generate a DRC file, click Yes. Any project that uses its own resource strings (instead of previously compiled RC files) needs a DRC file.
 - If you are sure that no new DRC files are needed (because your project doesn't introduce any resource strings of its own), you can select the "Skip DRC files that are not found" check box in the final dialog. This prevents the wizard from generating, or asking to generate, DRC files.Later, you can add languages to and [remove languages](#) from your project by rerunning the Resource DLL wizard.

The Resource DLL wizard creates a separate DLL project for each language you have selected. If Automatic Repository Query is checked on the Translation Tools page of the Environment Options dialog (Tools|Environment Options), the wizard populates resource DLLs with translations for any strings that have matches in the Repository. Otherwise, the generated projects contain untranslated copies of the resource strings in your original project. Your next step is to translate these strings in the [Translation Manager](#), and build your DLLs.

Because all the generated projects have the same name, they must be kept in separate directories. Do not attempt to edit the generated project (.DPR) files, since your changes will be overwritten the next time the DLL is updated.

References to the generated projects are maintained in the parent project's options (.DOF) file. The Resource DLL wizard also creates a project group that includes the parent project and each of the generated DLL projects. It's a good idea to save and use this project group, since it simplifies the task of recompiling the resource DLLs.

Updating resource DLLs

[See also](#)

To update all the resource DLLs included in your project,

1. Save and build your project.
2. Choose Project|Languages|Update Resource DLLs.

The Update command adds resources to your DLL projects, including new resources that were not in the parent project when the Resource DLL wizard was last run and resources that have been changed. It does not compile the DLLs. After updating, you need to rebuild each resource DLL project by opening the project in the IDE and choosing Project|Compile. To simplify this process, you can maintain all the resource DLL projects, along with the application itself, in a single project group that can be compiled from the Project Manager by choosing Project|Compile All.—

Removing languages from a project

[See also](#)

To remove languages from your project,

1. Save and build your project.
2. Select Project|Languages|Remove.
3. Check the languages that you want to remove. Click Next.
4. Click Finish.

The Resource DLL wizard removes the selected resource DLLs from your project's DOF file, but it does not delete the DLLs or the directories they reside in.

Restoring languages to a project

You can restore a language to your project after it's been removed.

1. Choose Project|Languages|Add to start the Resource DLL wizard.
2. Specify the directory path of the old resource DLL in the appropriate dialog.
3. In the Update Mode column, select Update.

For more information, see [Adding languages to a project](#).

Setting the active language

[See also](#)

The active language determines which version of your application runs in debug mode when you click the Run button or choose Run|Run. Before changing the active language, make sure you've recompiled the resource DLL for the language you want to use. Then,

1. Choose Project|Languages|Set Active.
2. Select a language from the list.
3. Click Finish.

To set the active language back to the base language you are developing in, choose Project|Languages|Set Active and select <none> at the top of the list.

Note Ordinarily, the primary (base) language in which you develop your application matches your computer's Windows system locale. If you have created a resource DLL for this language, your application's initialization code will try to load the DLL even when the active language is set to <none>. In this case, you can run the application without loading a resource DLL by temporarily moving or renaming the base-language DLL. For more information, see [Using resource DLLs](#).

Using the Translation Manager

[See also](#)

Before you can use the Translation Manager, you must [add languages](#) to your project with the Resource DLL wizard.

To open the Translation Manager, choose View|Translation Manager.

The Translation Manager is divided into two panes.

- The left pane is a tree diagram that displays the languages available in your project. (To display all available languages, make sure that the active project—the one shown in **bold** in the Project Manager—is your main application project, not one of the resource DLLs.) At the top of the pane are six speedbuttons:

- [Launch Repository](#) is equivalent to Tools|Translation Repository.
- [Add a Language](#) is equivalent to Project|Languages|Add.
- [Remove a Language](#) is equivalent to Project|Languages|Remove.
- [Set the Active Language](#) is equivalent to Project|Languages|Set Active.
- [Update](#) is equivalent to Project|Languages|Update Resource DLLs.
- [Exit](#) closes the Translation Manager.
- The right pane displays a grid for viewing and editing translations.

When you select a file in the tree diagram on the left, the Translation Manager displays the associated resources on the right. To switch focus between the left and right panes, press F6.

- The Id column shows the name of each resource in its form (.DFM) or resource file. Items preceded by ○: are objects defined in a .DFM file; items preceded by I: are inherited from another form; items preceded by L: come from a frame.
 - The first set of columns shows values in your source or *base* language (the primary language in which you are developing the application). The second set of columns shows corresponding values in the target language selected in the tree diagram. The columns further to the right (for example, “Previous English” or “Previous French”) allow you to track earlier versions of the source strings and their translations.
 - The Status, Comment, and date columns help you to keep track of your work.
 - The date columns—Created and Modified—are maintained automatically.
 - Status is automatically changed from Untranslated to Translated the first time you put text in a resource’s translation field; you can also toggle the Status field manually. When a resource has been removed from your project, it still appears in the Translation Manager, but its Status is changed to Unused. If the Status field shows Auto-Translated, it means that the translation was copied automatically from the [Repository](#).
- You can move and resize these columns with the mouse. To resort the list, click the top of the column for the field you want to sort on. Editing and other functions—such as filtering the list, hiding columns, and changing fonts—are available through the Actions speedbutton at the far right above the grid; you can also access this menu by right-clicking on the grid.

Note If your base and target languages use different character sets, you’ll need to change the font in the target-language column. To select fonts on a language-by-language basis, choose Tools|Environment Options|Translation Tools from the main Delphi menu, or right-click in the Translation Manager grid and choose the Font menu.

The Translation Manager displays only [Localizable](#) properties and their subproperties.

Properties represented in the Translation Manager can be of various types, including integers, strings, bitmaps, helper classes (like TFont), and enumerated types (like TBiDiMode). Hence, in addition to translating text, you can use the Translation Manager to adjust other properties—such as fonts or the sizes of visual components—in localized versions of your application. **When translating string properties, always enclose the strings in quotation marks**; form (.DFM) files use single quotation marks, while Windows .RC files use double quotation marks.

The only editable fields in the Translation Manager grid are Status, target language, and Comment. After editing in the target language column, you need to recompile the resource DLL.

[Editing in the Translation Manager](#)

- [Using the Multiline editor](#)
- [Keyboard shortcuts](#)

Using the Repository

Using the Auto Translate option

Editing in the Translation Manager

[See also](#)

There are three ways to edit in the Translation Manager:

1. Edit directly in the grid. For target-language and Comment fields, click on the value you want to edit and type directly into the field. For Status fields, click on the field, then use the drop-down pick-list.
2. Use an Edit dialog. Right-click on any row or rows in the grid (but *not* in an editable field), select Edit, then select the field you want to modify. In the dialog, enter a new value for the field. You can select and edit several rows at once, provided the rows contain data of the same type.
3. For translations with more than one line, use the Multiline editor, described below.

Remember to update and rebuild your DLLs after editing.

Using the Multiline editor

The Multiline editor allows you to edit translations that are lengthy or contain multiple lines of text separated by hard returns. To start the editor, place the cursor on any row in the grid, then click the Multiline Editor speedbutton (on the right above the grid). The editor displays the source and target languages in separate panes. Only the target language is editable.

A row of speedbuttons at the top of the editor provide navigation and other functions.

- Use the Up and Down arrows to move to the previous or next string in the Translation Manager grid.
- Use the Tile Across and Tile Down buttons to change the orientation of the editor panes.
- Click OK (the check-mark button) to save your changes.
- Click Cancel (the X button) to discard unsaved changes and close the editor.
- Use the Font button to change the display font in the editor. The changes apply to whichever pane the cursor is in when you click the Font button and affect the Translation Manager grid as well as the Multiline editor.

Keyboard shortcuts

The following keyboard shortcuts are available in the Translation Manager. You can also perform these actions by right-clicking and choosing from the pop-up menu.

Ctrl+N	Find next untranslated item.
Ctrl+O	Copy text from source (base language) column into translation column.
Ctrl+T	Change status to Translated.
Ctrl+P	Restore translation from Previous translation column.
Ctrl+D	Copy translation from prior row in grid.
Ctrl+A	Select everything in grid.

Using the Repository

[See also](#)

You can access the [Translation Repository](#) from the Translation Manager grid by right-clicking on any resource and choosing Repository. From the Repository menu, choose **Add Strings to Repository** or **Get Strings from Repository**.

- **Add Strings to Repository** stores the translation from the selected row(s) into the Repository database.
- **Get Strings from Repository** searches the Repository for a translation in the target language whose source string matches the selected resource.

If you choose **Get Strings from Repository** and the Repository contains exactly one translation that matches the selected source string, it copies that translation into the target-language column. If the Repository contains more than one match for the selected resource, its default behavior is to retrieve the first matching translation it finds; you can change this behavior by choosing Tools|Environment Options|Translation Tools from the main Delphi menu, then selecting an option under Multiple Find Action:

- Select **Skip** if you want the Translation Manager to retrieve nothing from the repository when it finds more than one match.
- Select **Use first** (the default) if you want the Translation Manager to retrieve the first matching translation in the Repository.
- Select **Display selection** if you want the Translation Manager to offer you a choice when it finds multiple matching translations.

Remember to rebuild your DLLs after editing.

Using the Auto Translate option

To enable the Auto Translate option, choose Tools|Environment Options|Translation Tools from the main Delphi menu, then check Automatic Repository Query. Each time your DLLs are updated, Auto Translate searches the Repository for translations whose source values match the ones in the Translation Manager grid.

If Auto Translate finds exactly one match for a string, it retrieves that translation and copies it to the DLL project. If it finds more than one match, its default behavior is to retrieve the first matching translation in the Repository. You can change this behavior by choosing Tools|Environment Options|Translation Tools from the main Delphi menu and selecting **Skip** or **Display Selection** under Multiple Find Action. For the Auto Translate option, **Display Selection** is equivalent to **Skip**; if either of these options is selected, Auto Translate skips strings for which it finds more than one match.

When a translation has been copied automatically from the Repository, it is marked as Auto-Translated in the Translation Manager's Status column.

Setting up a translation repository

[See also](#)

While working in the Translation Manager, you can [use the Repository](#) to store and retrieve translated strings. You can also access the Repository directly, through its own interface, to perform such tasks as importing and exporting data, searching for specific strings, and deleting unwanted translations.

To open the Repository, choose Tools|Translation Repository or click the Launch Repository speedbutton in the upper left corner of Translation Manager.

The Repository stores data, in binary format, in a file with the .RPS extension. By default it uses DEFAULT.RPS in the BIN directory.

- To save the current data in a different file, click the Save As button.
- To open a different RPS file, click the Open button.
- To open a new, empty RPS file, click the New button.
- To search for a string in the current repository column, click the Find button.
- To delete a record, select any row and press the **Delete** key.

Several Repository settings are controlled from the Translation Tools page of the Environment Options dialog. To change these settings, choose Tools|Environment Options from the main Delphi menu and click the Translation Tools tab.

- To select a different RPS file, enter a file name under Repository.
- To make the Resource DLL wizard automatically retrieve strings from the Repository, select Automatic Repository Query.
- To determine how the Repository responds when it finds more than one translation for the same source string, select an option under Multiple Find Action.

Importing and exporting data

The Repository's Import and Export functions allow you to exchange data between RPS files. This is useful for sharing translations in workgroups and development teams. Importing and exporting are accomplished through intermediate text files in XML format.

- To Export the contents of the current repository to an XML file, click the Export speedbutton. When the file-selection dialog opens, specify a file name and click Save.
- To import the contents of an XML file into the current repository, click the Import speedbutton. When the file-selection dialog opens, choose a file and click Open.

Note The Import and Export functions require MSXML.DLL. If you don't have this file on your system, you can get it by installing Microsoft Internet Explorer 5.

ITE-generated files

[See also](#)

Files generated by the ITE include

- .DFN files. The ITE maintains a separate .DFN file for each form in your application and each target language. These files contain the data (including translated strings) that you see in the [Translation Manager](#).
- .RC files. The [Resource DLL wizard](#), using the compiler-generated .DRC file, creates an .RC file for each target language. These .RC files contain special comments that are used by the ITE.
- .RPS files. The [Translation Repository](#) stores data in an .RPS file. You can maintain more than one repository by saving multiple .RPS files.

You should not edit any of these files manually.

Find dialog

[See also](#)

Use the Find dialog to search for text in the Translation Manager.

- Enter the text you want to search for after Text to Find.
- Click the Font button (to the right of Text to Find) to change fonts.
- Under Options, you can enable case-sensitivity and whole-word search. Select Ignore Pick Letters to disregard the ampersands (&) that indicate accelerator keys in button and menu-item captions.
- Under columns, select the columns you want to search. (To select or unselect all columns, right-click and choose Check All or Uncheck All.)
- Under Scope, select Global to search all rows or Selected Rows to search only the rows highlighted in the translation manager.

Find in Files dialog

[See also](#)

Use the Find in Files dialog to search multiple files in the Translation Manager.

- Enter the text you want to search for after Text to Find.
- Click the Font button (to the right of Text to Find) to select a font.
- Under Options, you can enable case-sensitivity and whole-word search. Select Ignore Pick Letters to disregard the ampersands (&) that indicate accelerator keys in button and menu-item captions.
- Select Search Files in Project to search only the currently open project. Select Search in All Projects to search all projects in the current project group.
- Under File Types, select Forms, Resource Scripts, or both. Then choose the columns you want to search from the list on the right. (To select or unselect all columns, right-click and choose Check All or Uncheck All.)
- You can select different search columns for each file type. Highlight Forms or Resource Scripts, then select search columns.
- To use the same search columns for Forms and Resource Scripts, select Keep Check Lists Synchronized.

Select a Font (Translation Manager Find-in-Files dialog)

[See also](#)

Use this dialog to specify a font for the text you enter in the Text to Find field. The dialog displays all ITE fonts used in your project. (To configure ITE fonts, choose Tools|Environment Options|Translation Tools.)

