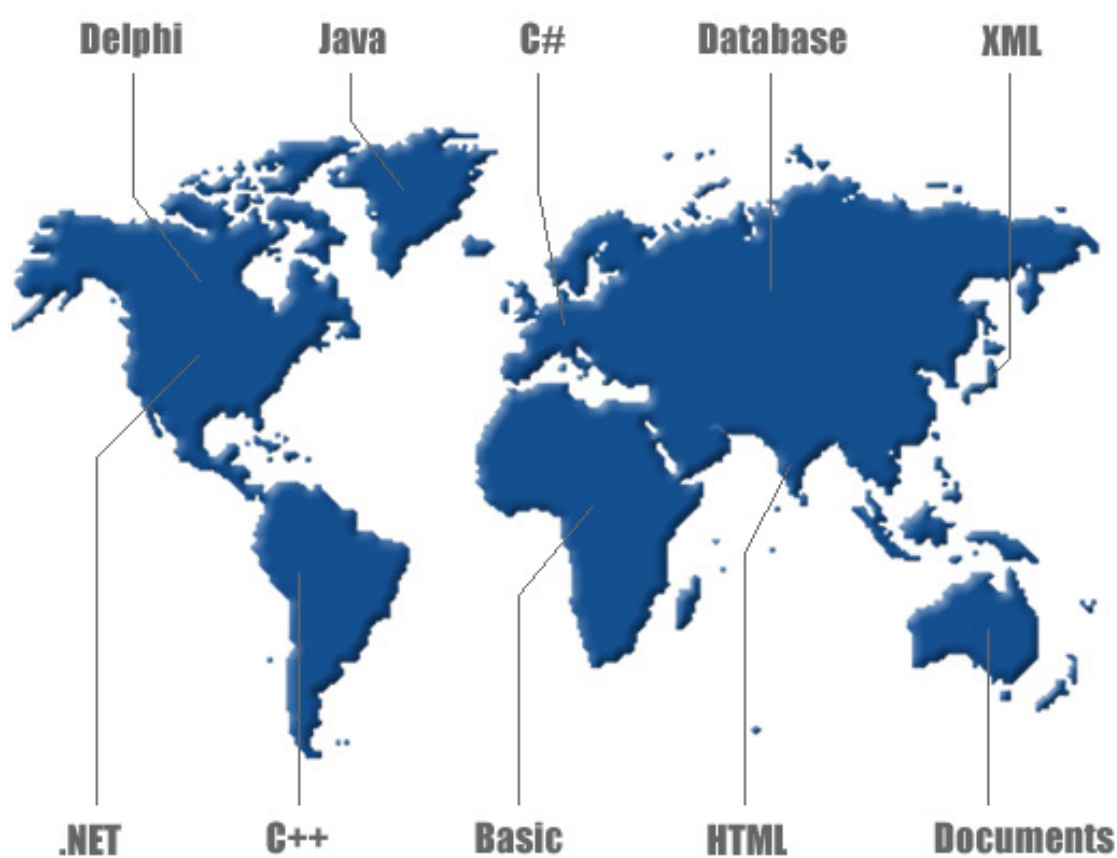




Three simple steps to localize



Quick Start Guide

Copyright © 2006-2011 Sisulizer Ltd. & Co KG

Content changes reserved.

All rights reserved, especially the permission to copy, distribute and translate this document.

Reproduction, processing, duplication and distribution require exclusive permission from Sisulizer Ltd. & Co KG.

All information in this document is reviewed in all conscience. Sisulizer Ltd & Co KG will not be responsible for any damage caused by using this documentation.

Sisulizer Ltd & Co KG

Kirchstr. 37

52156 Monschau

Email: sales@sisulizer.com

<http://www.sisulizer.com/>

All trademarks are registered trademarks of the corresponding vendors.

Sisulizer Ltd & Co KG explicitly respects the rights of these vendors.

Document version 3 for Sisulizer version 3

Table of contents

0.	About this manual	4
1.	Scan	5
1.1.	Create a new project	6
1.2.	Distribute the work	14
1.3.	Exchange your project with Sisulizer's help	14
1.4.	Switching from any other localization tool	20
1.5.	Localizing upgrades of your software	25
2.	Translate	28
2.1.	Open the project	29
2.2.	Start to translate (the translation grid)	34
2.2.1.	The translation sheet	35
2.2.2.	WYSIWYG – visual translation	37
2.2.3.	Editing the translation	37
2.3.	Translation status	38
2.3.1.	Not translated	38
2.3.2.	Auto translated	39
2.3.3.	Why translated is not the same as complete	39
2.4.	Getting an overview: Filter	39
2.5.	Comments	41
2.6.	Check your progress (statistics and reports)	41
2.7.	Reports	42
2.8.	Automatic validation	44
2.9.	Finish your work	47
3.	Build	49
3.1.	Put the pieces together (merge translations into your project)	50
3.2.	Check the progress (statistics and reports)	54
3.3.	Automatic validation	55
3.4.	Selecting the output format	60
3.5.	Create the localized version of your software	61

0. About this manual

The purpose of this guide is to give you an overview about the functions and processes you will use most often when you work with Sisulizer. It will give you a quick head start into all functions you need to start localizing right away.

This guide is intended to show you software localization. While it is perfectly possible to localize other things with Sisulizer (amongst them XML, HTML or databases), this guide will focus solely at software localization.

This manual is divided into three sections, where each fits its own audience:

Scan – Read this if you are a developer

page 5



Read this section if you have access to the latest binaries or source code of the software you want to localize. This usually applies to developers, project managers or simply the people in charge of the localization project.

Translate – Users of all editions should read this

page 28



Read this section if you are a translator and only need to know how you can actually translate something with Sisulizer. You don't need any programming experience or access to the latest binaries or source code. If your task is to do the translation (e.g. with the Free or Translator Edition), then this section is for you.

Build – Read this if you are a developers or user of the Translator Edition

page 49



Read this section if you are the person who puts all the pieces together. You will gather the translated parts from your translators and build the ready-to-use localized application.



1. Scan

If you use the Free or Translator Edition you can skip this chapter.

This section is intended for those who will start the localization process. This will be for example developers or project maintainers. You have finished a software project in one language and you want to start the localization process now.

You will learn about the following topics:

- Create a new project (page 6)
This topic will show you how to create a new Sisulizer project from scratch. All you need is a current build of the software you want to localize (and some specific files depending on your development environment).
- Distribute the work (page 14)
Now it's time to have your software translated. You probably want to have this work done by professional translators. But you don't want to send them just everything - why should they translate fixed system messages or see languages they are not supposed to translate?

- Exchange your project with Sisulizer's help (page 14)
You will learn how to distribute your resources to your translators without giving away any internal information.
- Switching from other localization tools (page 20)
If you have used other localization tools before, it's very easy to switch to the comfort of Sisulizer. Learn about how to switch from other tools easily.
- Localizing updates of your software (page 25)
You have created an update for your software, which is already localized with Sisulizer. Learn how to rescan your project and reuse all existing translations in Sisulizer.

1.1. Create a new project

Install Sisulizer:

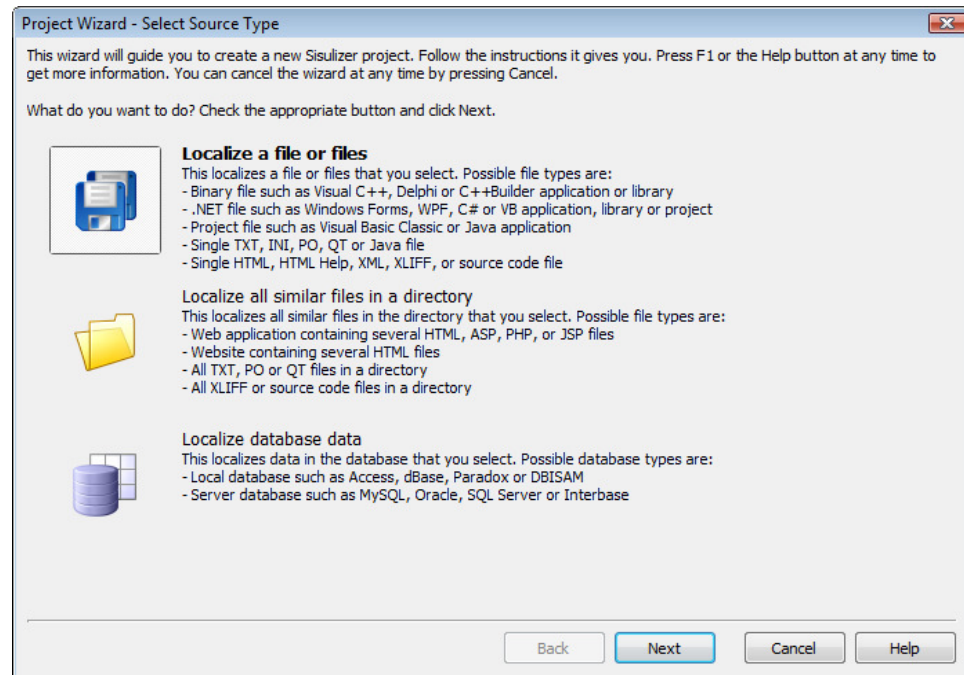
Simply download the installer from <http://www.sisulizer.com/downloads.shtml> and follow the instructions. Then start the application.

The converter sample coming with Sisulizer is needed to repeat the steps in this manual. So please make sure that you choose to install the samples coming with Sisulizer.

At first start, Sisulizer shows a simple interface which is empty for the most part.



Find the New Project button to open the Project Wizard.

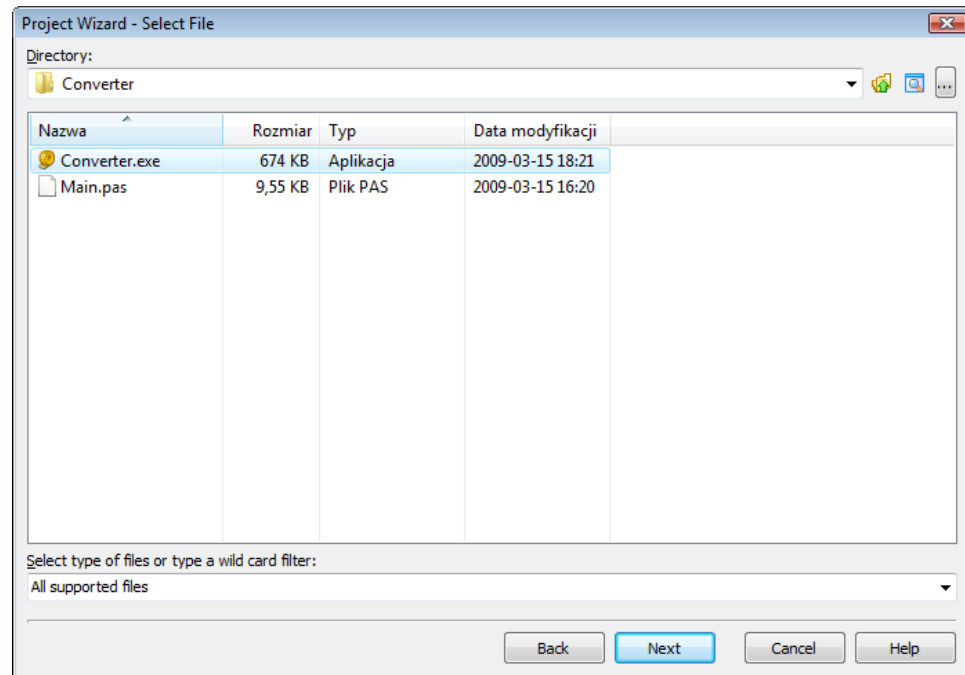


The first step in Project Wizard is to select which type of source you want to localize.

There are three options:

- Localize a file or files
- Localize all similar files in a directory
- Localize database data

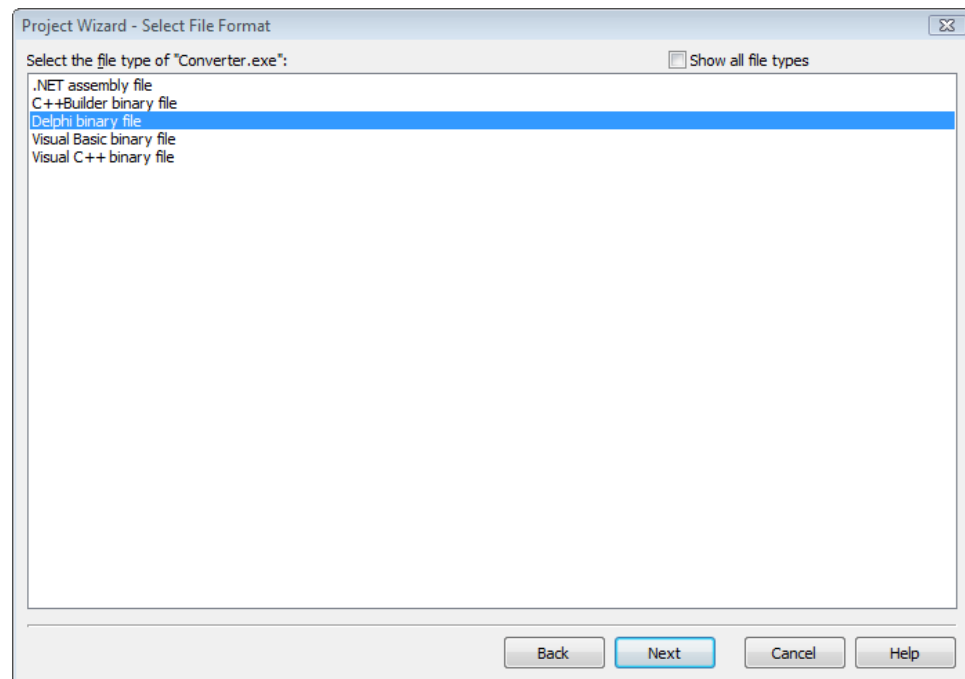
We will assume you have one executable to localize. As an example we will use the Converter.exe coming as sample with Sisulizer. The button Localize a file or files is already selected, so you'll only have to click on `Next`.



In this step you select the application to localize. Browse to the location of `converter.exe`. `Converter.exe` is located in `<your Sisulizer directory>\VCL\Delphi\Converter\` (XP) or `\Users\<user folder>\Documents\Sisulizer 3\VCL\Delphi\Converter\` (Vista or newer).

Select `converter.exe` from the list and click on Next to proceed.

Sisulizer will now determine what type of application the selected file is. In case of `Converter.exe` this will be `Delphi binary file` and the corresponding entry will be already selected.



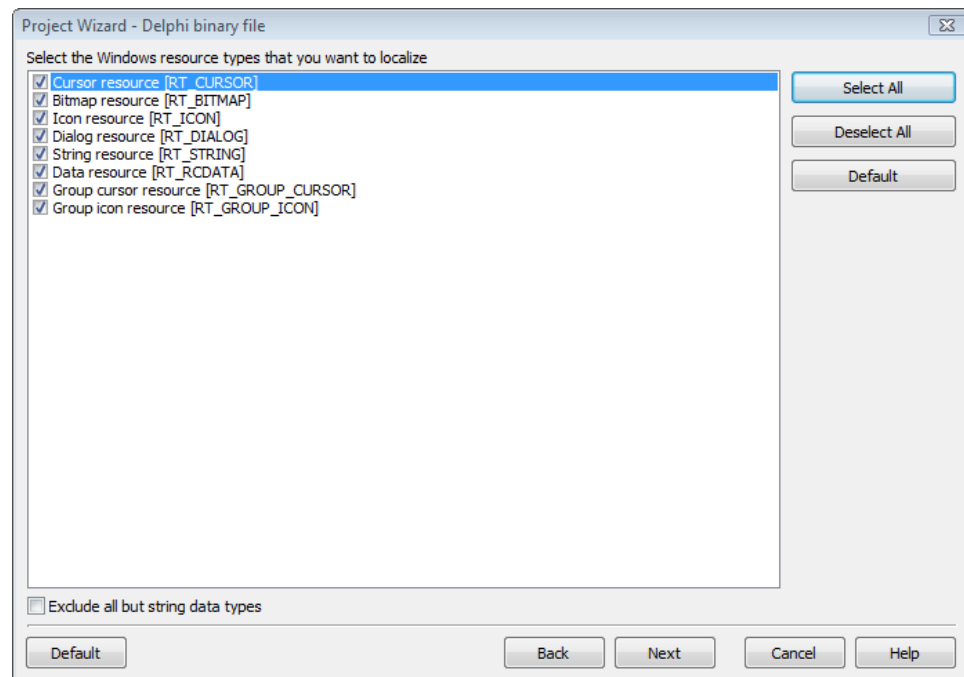
You can select one of the five binary formats supported by Sisulizer. To see a complete list of formats (including text formats), enable the checkbox `Show all file types`.

Sisulizer tries to find the appropriate file format of your application and selects it automatically. Usually, it does a good job finding the correct file type. If not, you can select it here manually.

Click `Next`.

Delphi binaries usually consist of many different resource types. There are, for example, Icon resources, Dialog resources and String resources, to name a few.

It is the developer's responsibility to put as much visible information as possible into resources. This is the common way for Windows applications and makes them easier to localize.



Sisulizer selected all available resources for you. This is the default for a complete localization.

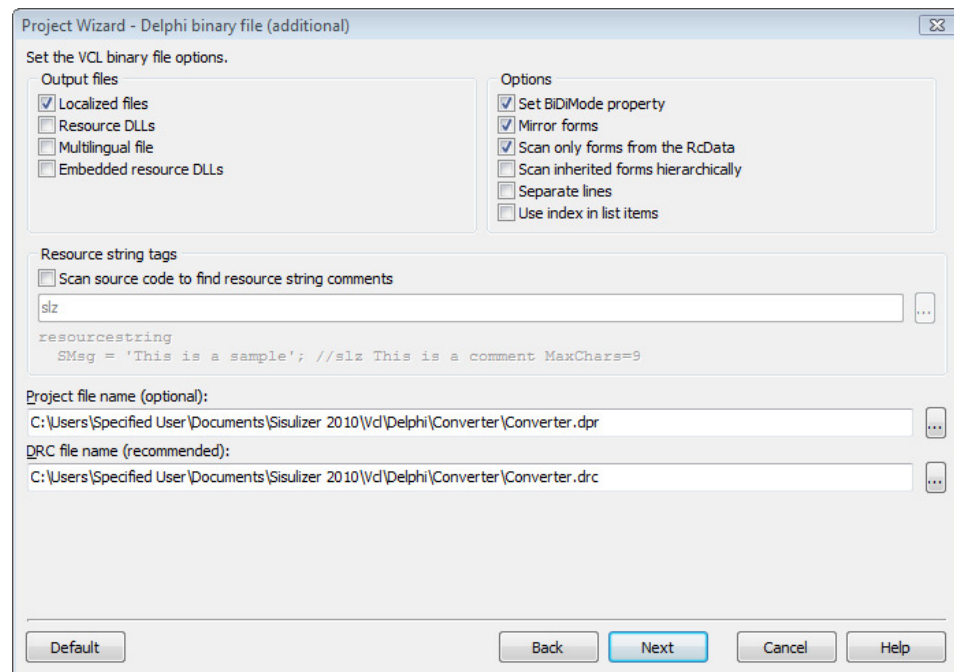
If you only want to localize strings, click on `Default` or check `Exclude all but string data types`.

Just in case, you're wondering. Yes, you can localize images, icons and other binary data with Sisulizer.

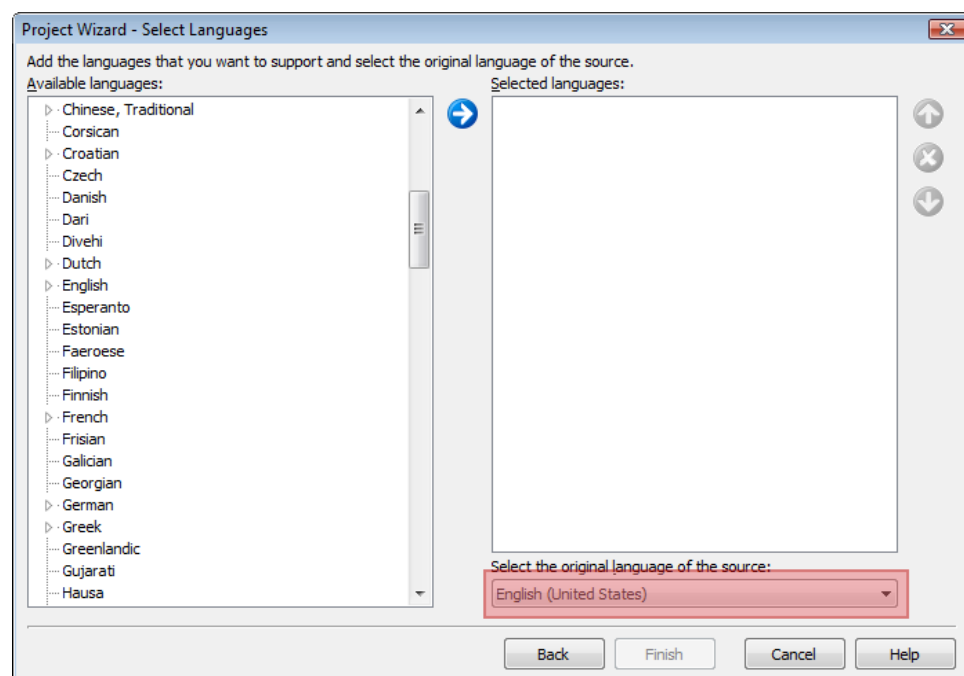
Please select all here; we want to show you Sisulizer's WYSIWYG ability. Click `Next` to proceed.

The following page of the wizard allows you to give the wizard additional information of your project. The most important in case of Delphi is the `DRC file name`. The wizard tries to find this file automatically for you. The Converter sample comes with a matching `.drc` file and the wizard enters its name for you. All other setting are correct for our sample and you can click `Next` to proceed.

If you are using Delphi and want to learn why the `.drc` file is important for successful Delphi localization you should read our document "Localizing Delphi and C++Builder applications". It can be downloaded from the download page of www.sisulizer.com.



The next step is to select the language of the original application, as well as the languages you want to translate to ("target language").



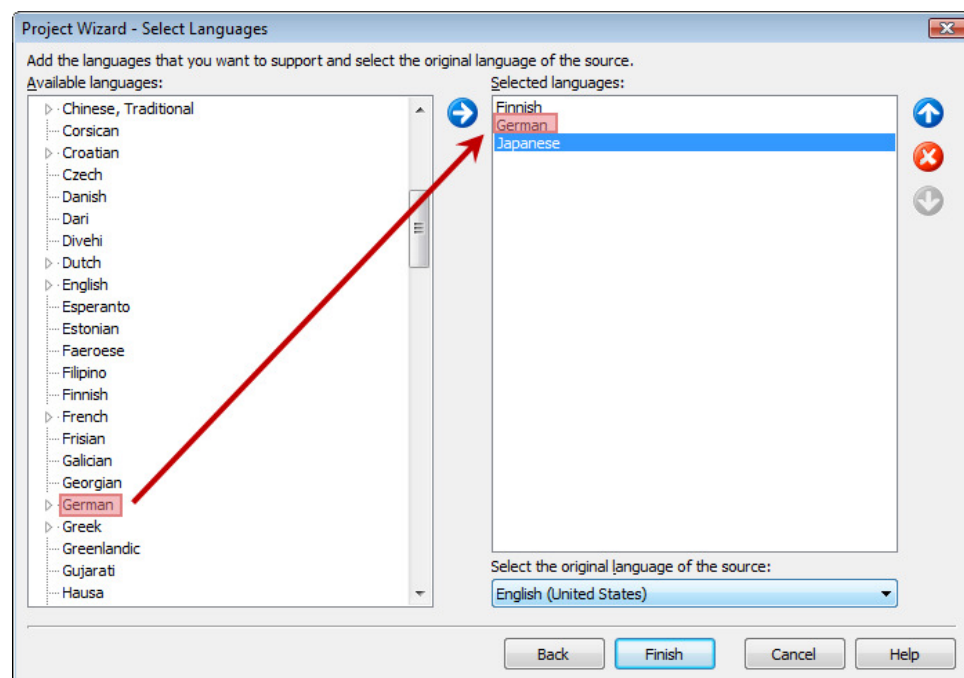
Sisulizer has usually found the original language at this point.

The highlighted region on the screenshot shows the original language selection.

Have a closer look at this language as you cannot change it afterwards. Make sure, Sisulizer has really found the original language of your application

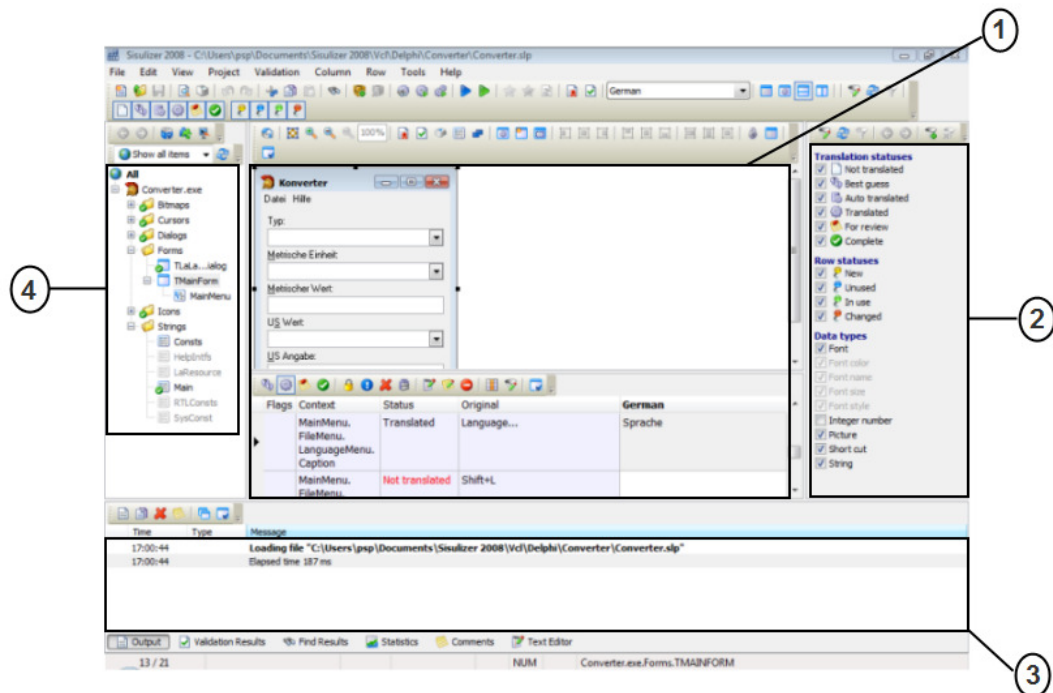
Now you have to select at least one language to translate to. You may select as many languages as you like and also variants of languages like English (US) and English (UK).

Don't worry, you can edit the supported languages afterwards and add new or remove old ones.



To add a language to your project, just drag and drop an entry from the list on the left to the right. Simple as that!

It is a good idea to save your project now. If you followed our sample you will find that there is already a file called "converter.slp" coming with Sisulizer. So it is a good idea to save your work as "converter1.slp" to not overwrite the original Sisulizer sample project.



There are some language-entries with plus signs to their left. These languages contain definitions for sublanguages. If you'd like to add a specific translation for English (Zimbabwe), click on the plus to the left of `English` and drag the appropriate entry into the right list box.

If you have selected all the languages you want to support, you can let Sisulizer scan your application and extract your resources. To do so, click `Finish`.

After scanning, the user interface will be similar to the above screenshot. The four main parts are the following:

- ① The translation sheet and WYSIWYG display.

Here your translators do all their work. Sisulizer has WYSIWYG capabilities for all visual components like dialogs and menus.

- ② The filter pane.

You can quickly filter out certain view conditions using this pane.

- ③ The information pane.

You'll see error and success messages as well as statistics, comments and validation results.

- ④ The project tree.

As you have seen, software consists of several resources, like icons, strings, dialogs and more. Here you see an ordered view of resource types available.

Everything is set up to begin translating now. Read on to learn more about how you create a package for your translators.

If you'd rather like to translate yourself, jump right to the next section: Translate (page 28).

1.2. Distribute the work

By now you have created a new Sisulizer project. Sisulizer scanned your application, found resources and extracted them. Sisulizer now has all the information it needs in the project data. From now on, there is no need to have your application at hand; until the next update that is.

You can now start translating. Or, in most cases, you will have professional translators to take this task. You could give the entire project to your translators but it will certainly contain resources which they shall not translate or which they even should not see at all.

Remember, the translator will not see any sources. They get a complete representation of your software without ever seeing the source.

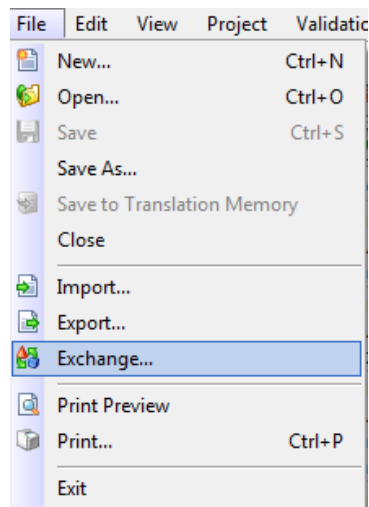
While Sisulizer can manage a lot of languages at once, maybe you only want to send one language to each translator. Also, things like icons may or may not be interesting to translate. In our example case, we don't give them along. We pretend, our translator cannot handle images, but only text.

We will create an example project of Converter which will consist of:

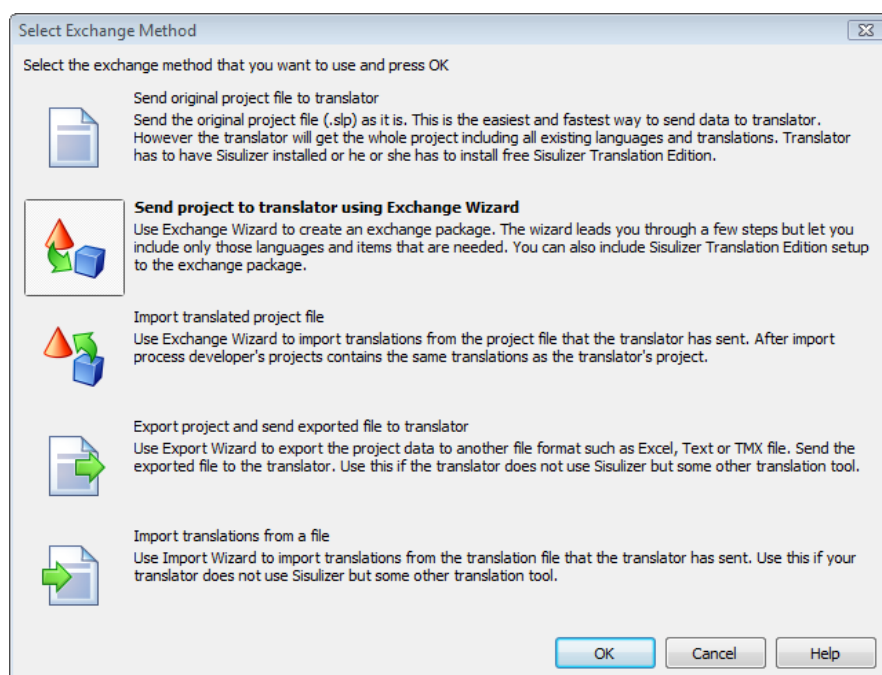
- the original language data and one single target language
- only dialogs, menus, strings and version resources
- a free of charge executable Sisulizer Free Edition for your translators

1.3. Exchange your project with Sisulizer's help

In the `File` menu, click on the entry `Exchange...`

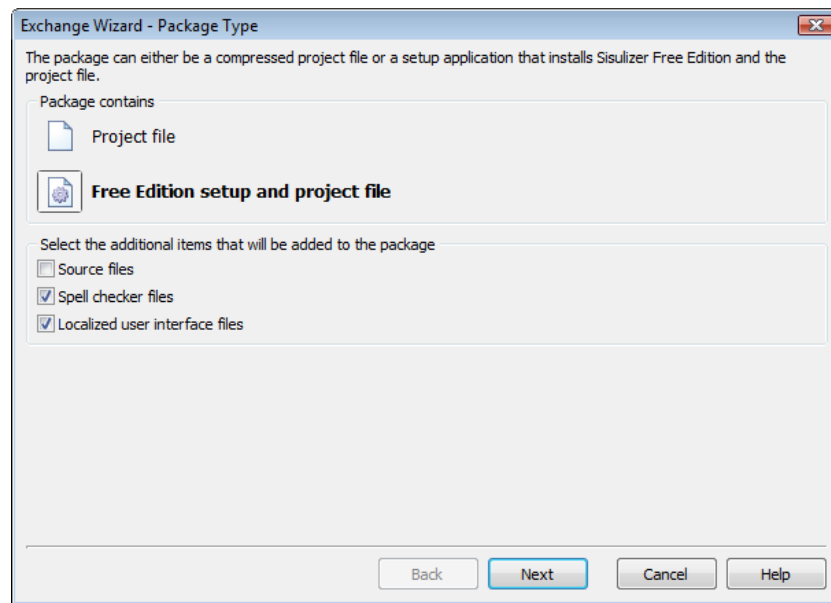


This will open Exchange Wizard. You can choose from five options. Here we describe the two options: Send project to translator using Exchange Wizard and Import translated project file.



Select Send project to translator using Exchange and then click OK to continue.

In Exchange wizard you select what kind of information you send to your translators.



In this step you choose that your project will include a Free Edition.

If your translator used Sisulizer before and so already has a copy of the Free Edition, select "Project File".

If this is the first time your translator receives a Sisulizer package or if you're unsure, select Free Edition setup and project file.

For our example, please choose Free Edition and project file. Leave the other two options as they are.

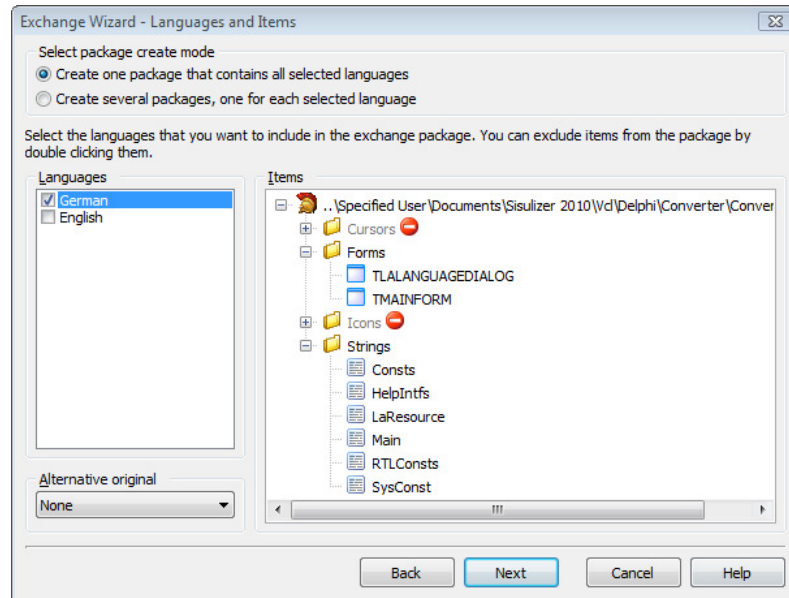
What is the Free Edition?

Free Edition is a free version of Sisulizer mainly used by translators. This version has visual editors (WYSIWYG) as well as everything else, Sisulizer brings along. There is also a Translator Edition sold in our shop. The Translator Edition is similar to the Free Edition but has much more features.

There are two main limitations, though: Free Edition can not create new projects and build localized versions nor scan/rescan a project. If your translators need to build the translated binaries you should take a look at Sisulizer's Translator Edition.


Click **Next** to continue.

In the next step you are asked which languages and resources you'd like to include in the package.



The page you see now is split into two parts. On the left you see the available languages.

This works as a standard Windows list box: Click on an entry to select only this entry, hold the `<Ctrl>` key and click to add another entry to the selection.

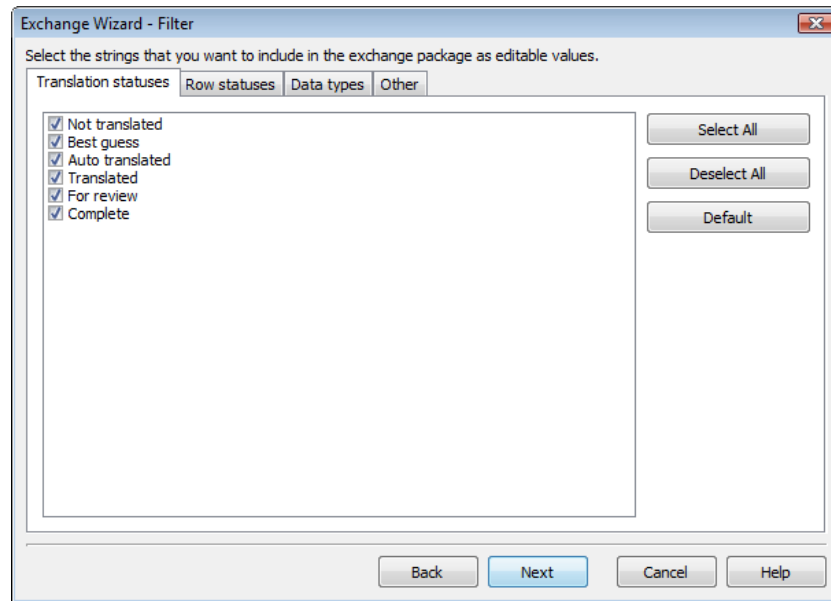
On the right you see a tree of resources. All of them are selected by default. To exclude a resource, double click on it. A symbol will appear next to the name: 

You can exclude single resources or whole categories. Please exclude anything but dialogs, menus, strings and versions now.

After you have made your selection on what to put in the package, click `Next` to continue.

Next you can select values with certain states to be in the package. If you've just created a project, everything will be of status "not translated" and "new".

Every translation has a status that you can use to order and categorize these translations. Learn more about states in the next topic of this section.

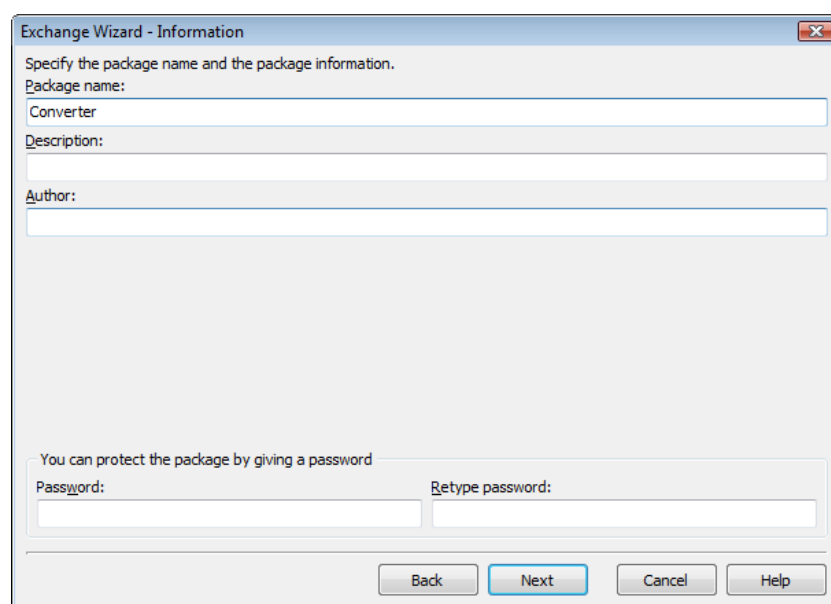


If you already translated parts, you could exclude them here by disabling the `Translated` option. Translators will not see anything you disable here.

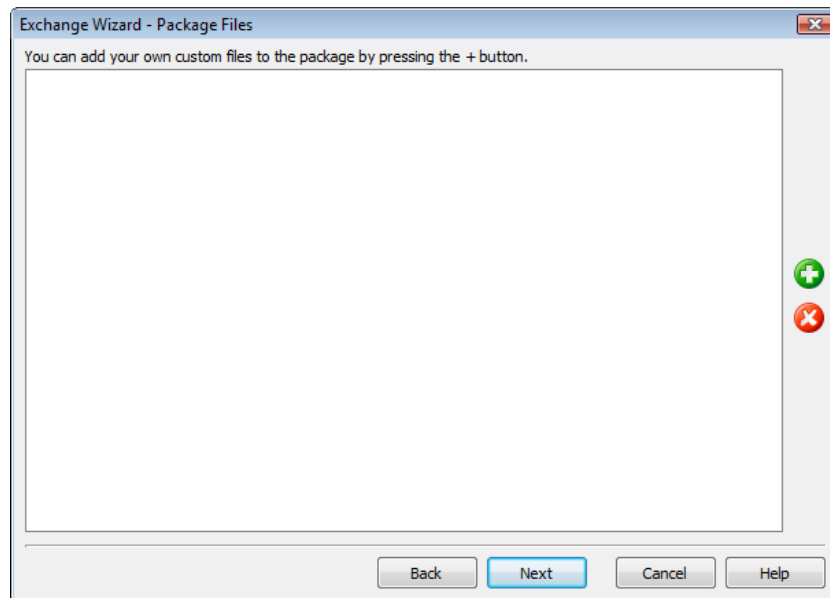
For now, it will be sufficient to let these options as they are. This will simply include everything. As nothing is translated yet, it doesn't really matter.

Click `Next`, we're almost finished.

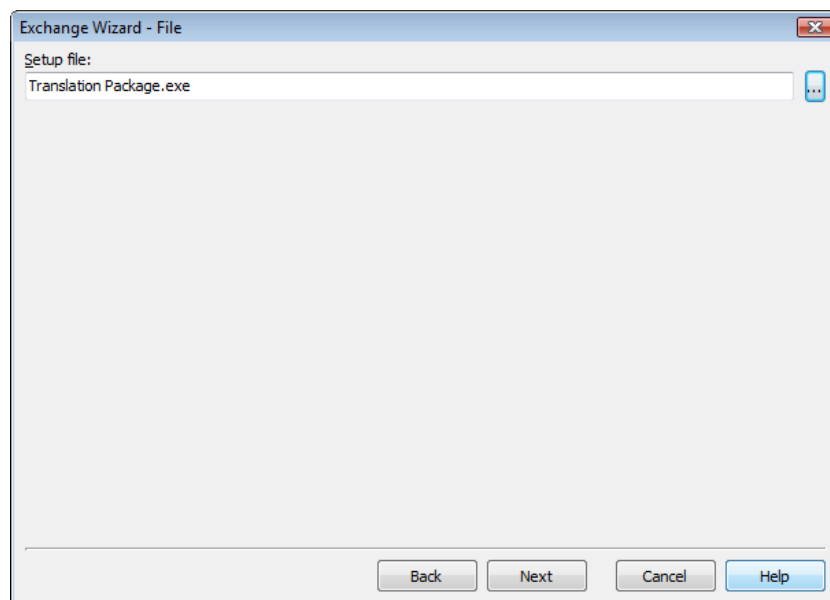
The next two pages will give you the opportunity to enter some information about the package you're sending, protect it by password and deliver some additional files.



For example you could add a `Readme` file which explains some conventions when translating your project.



Click `Next`, until you have to enter a filename:

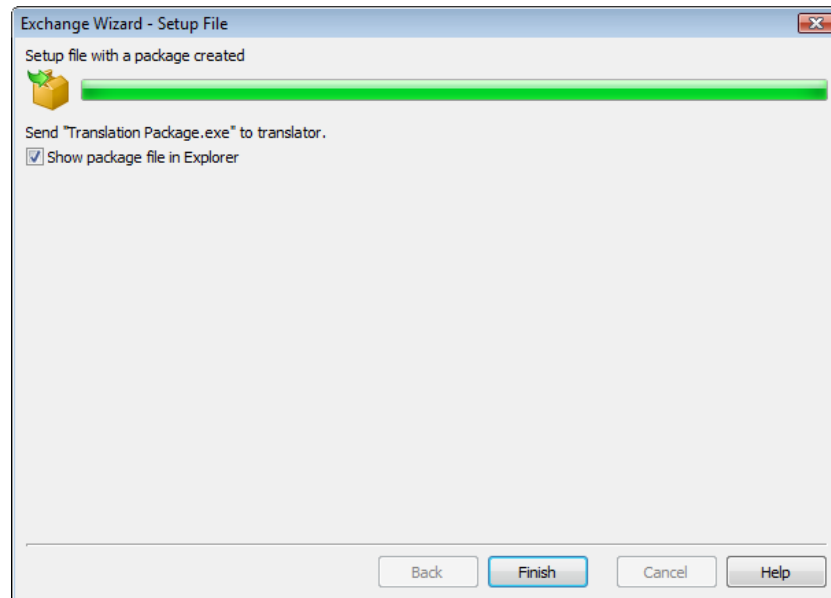


This is the last step. You have to enter a file name for the translation package.

Click the "... " button to browse to the path you wish to save to.

If you're ready, click `Next` and Sisulizer will create the package for you.

Sisulizer now creates the package file. You only have to wait until it's done.



You can send the package file to your translator. Check the option `Show package file in Explorer` to open an Explorer window with the file you just created.

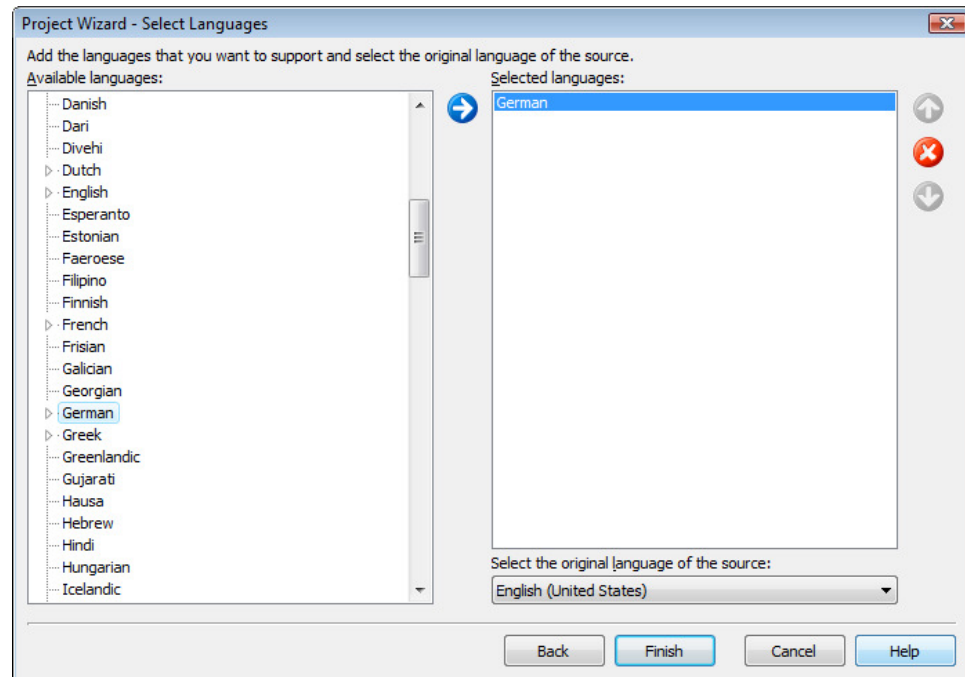
The translator gets a self-extracting archive including everything he needs. The section `Translate` contains information on how to use this package.

1.4. Switching from any other localization tool

Sisulizer makes it very easy to switch from another localization tool. In fact, this can be any localization tool as Sisulizer's method uses the binary output of these tools, namely your translated applications.

We will again take Converter as an example. Imagine you've localized Converter with another localization tool in the past. You translated from English to German.

Now create a new project with the English Converter as the source. When prompted to select target languages, select German:



You can also import translations from other sources. E.g. the Converter sample comes with Converter.slp containing German translations. This manual concentrates on the import from a localized binary only.

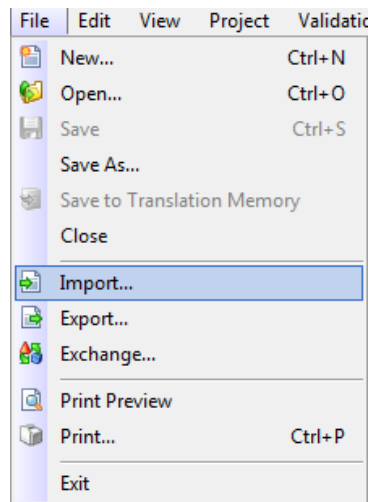
Now complete the project wizard and let Sisulizer scan the application.

You will now simply import the already translated application.

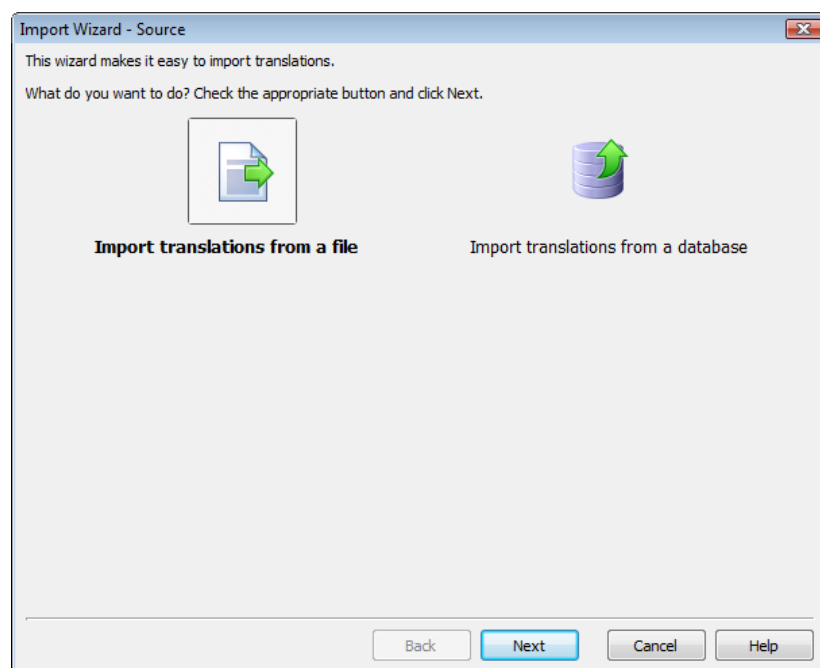
In our example, this is the German Converter.

If you want to translate more languages, add them all to the `Selected languages` list.

Open the file menu and click on `Import...`



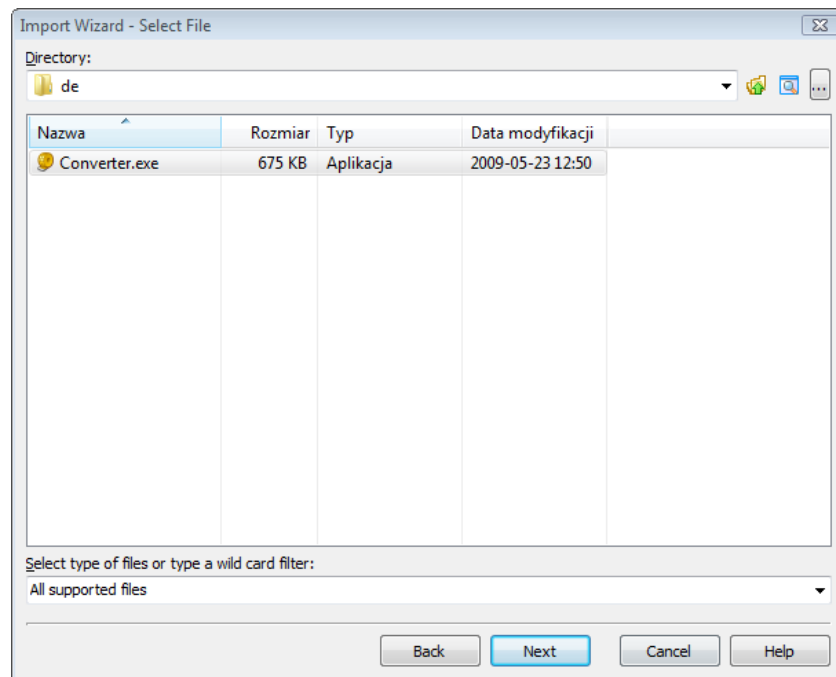
This brings up the import wizard:



Import Wizard prompts you to select an import source.

We use the translated application, so we select `Import translations from a file`.

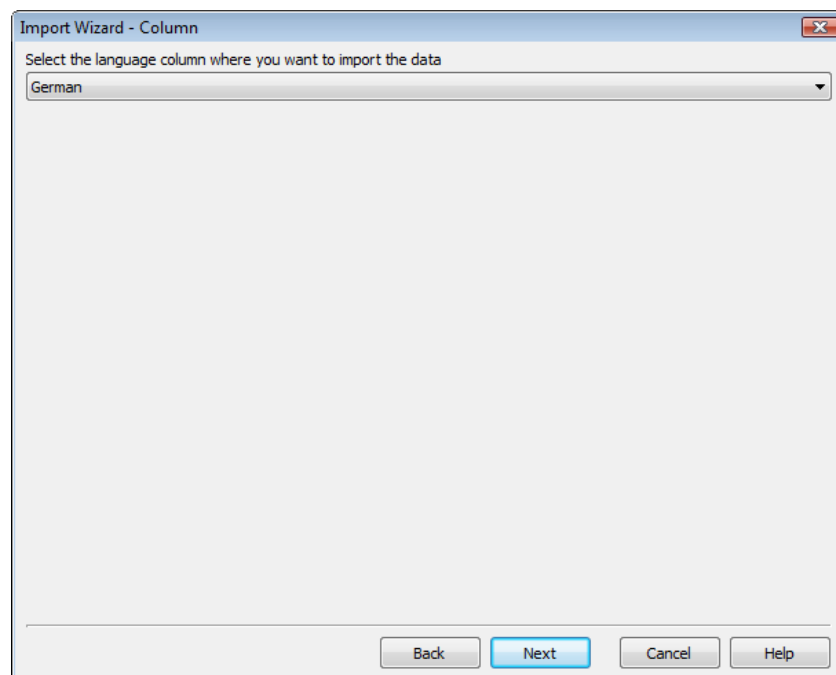
Click `Next` to continue.



Next, browse to the localized file of your other localization tool.

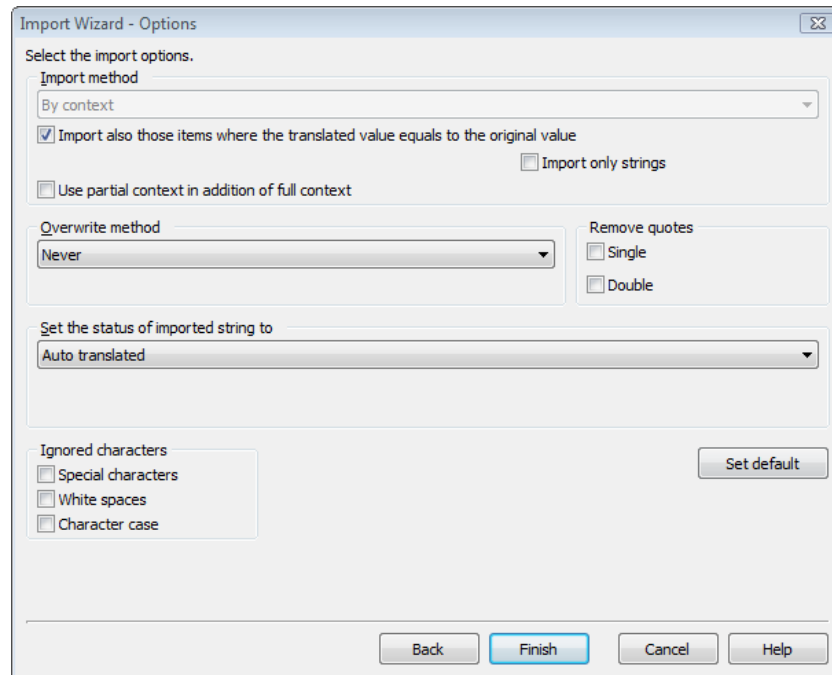
We select the German version of Converter.

Again, click **Next**. Sisulizer has found the target language:



Of course, you can select another language from the drop-down box if Sisulizer made a mistake.

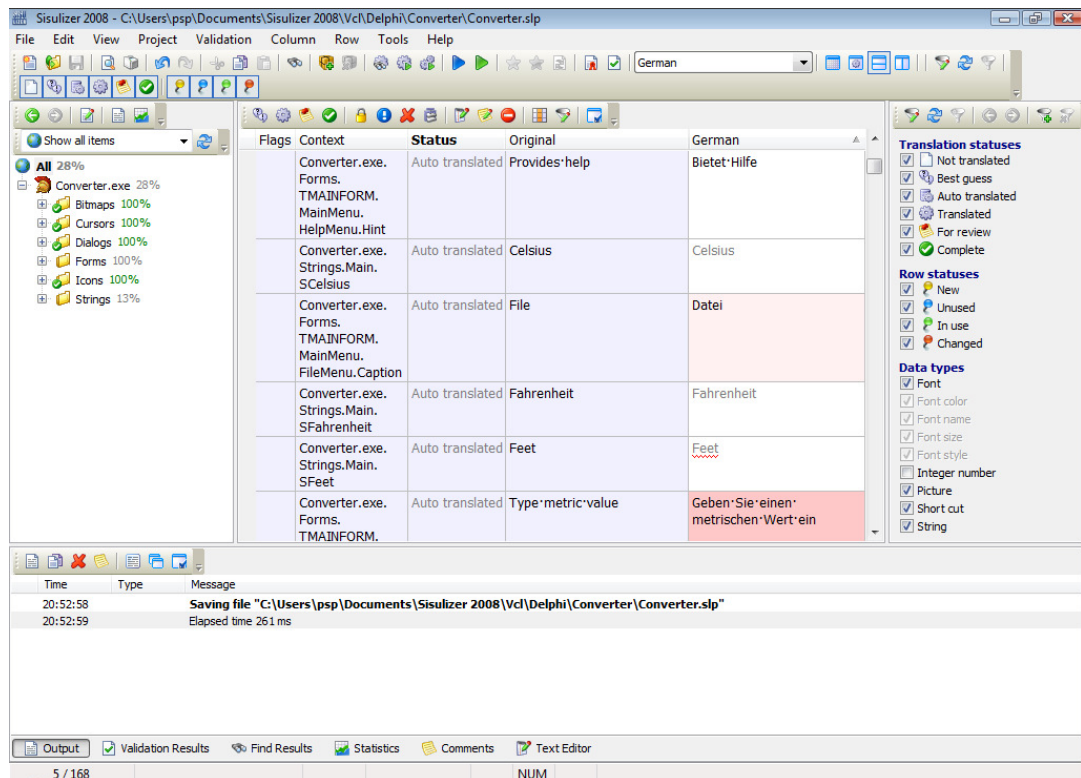
Click **Next** to continue:



This page will prompt for your preferred import method.

If you need detailed information on the options, simply press <F1> to open the online help of Sisulizer.

Click **Finish** and Sisulizer imports the localized strings:



Now you have a complete Sisulizer project already with all of the translations you've made with another localization tool.

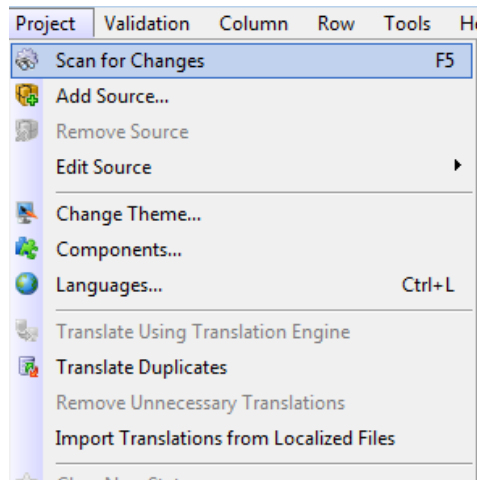
You can now localize upgrades of your application and enjoy the comfort of Sisulizer. Repeat the steps mentioned here if you want to import more languages. Read about localizing upgrades in the next topic.

1.5. Localizing upgrades of your software

Developing new versions of your software, you will usually want to have these new versions localized as well. But of course, you don't want to invent the wheel every time, so you want to only translate what has changed in the new version.

Sisulizer takes care of this. With only one click you can scan your application for changes.

To localize updates of your application with Sisulizer, simply click on **Scan** for changes in **Project** menu:



Sisulizer will notice what's already translated and will mark translations with different row statuses. You can see these row statuses from the translation sheet in the very first row. Colored bubbles represent the row status:

	Dialogs.14. 1200.Visible	Not translated	True
---	-----------------------------	----------------	------

There are four row statuses:

- new (yellow dot):
This will be strings which are new in your application. When you implement a new dialog box, the resources will become `new`.
- unused (blue dot):
Translations are `unused` if the software has changed and there is now a `new` translation. This is nice if you want to keep them as a reference.
- in use (no dot):
These are strings which are used in your application. Either from the former version or translated in the new version.
- changed (red dot):
If strings have changed in your application, but have the same context, Sisulizer will mark them as `changed`.

Usually, you only have to translate all the new strings and your upgraded application is localized.

If you've done this, continue with the steps described in the third section of this guide: Build.



2. Translate

This section is for the people who have the task of translating software but don't care about nifty technical details.

You will learn about the following topics:

- Open the project (page 29)
If you have just received a single file which you have to translate and you wonder what to do with it, this topic is for you. You will learn how to open the translation project in Sisulizer. You don't have to care about what Sisulizer is or if you have such a tool at all.
- Start to translate (the translation grid) (page 34)
Do you have a sheet in front of you with a lot of rows? Fine, you can start translating now!

You will learn how to use the project's nodes to divide this bunch of rows into a few at a time and how to visually check if your translations fit into dialog boxes, menus, message boxes and so on.
- Translation status (page 38)

Although Sisulizer helps to maintain contexts and produce adequate translations, there may be times when you're unsure if a translation exactly fits the original by context.

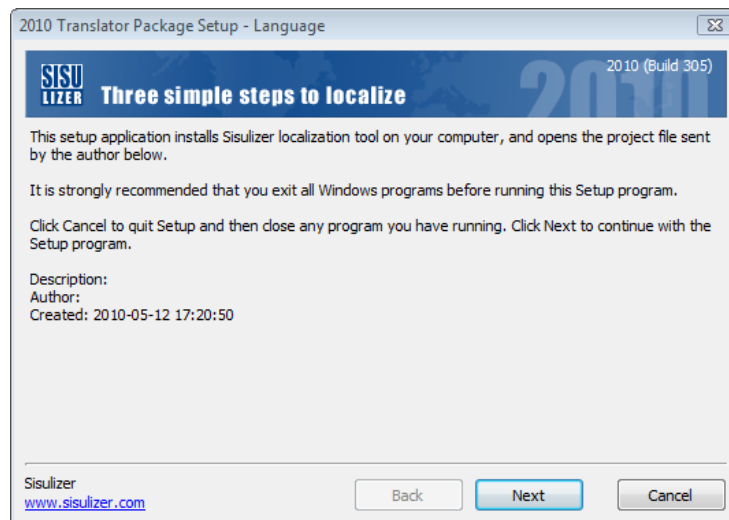
You will learn how to use translation states to mark them as "sure hit" or "not sure, better review this one".

- Getting on overview: Filter (page 39)
Filters help you to see only rows you want to see.
- Comments (page 41)
Comments are handy for the communication of translator and developers.
- Check your progress (statistics and reports) (page 41)
If you want to keep track of your translation progress you should have a look at the statistics panel. It shows how much you have translated and if parts are translated automatically.
- Reports (page 42)
You can even create a report and tell your boss you've already done 80% of the job for example.
- Automatic validation (page 44)
Also, you will learn about methods to automatically validate your translations and have minor flaws fixed by Sisulizer.
- Finish your work (page 47)
If you have just finished the translation with Sisulizer and want to send your work to you client, then read this topic. Here you will learn what your client needs and how to get that.

2.1. Open the project

If you received a Sisulizer project, there are two possible formats this project can be. You can determine by the file extension which format you're facing:

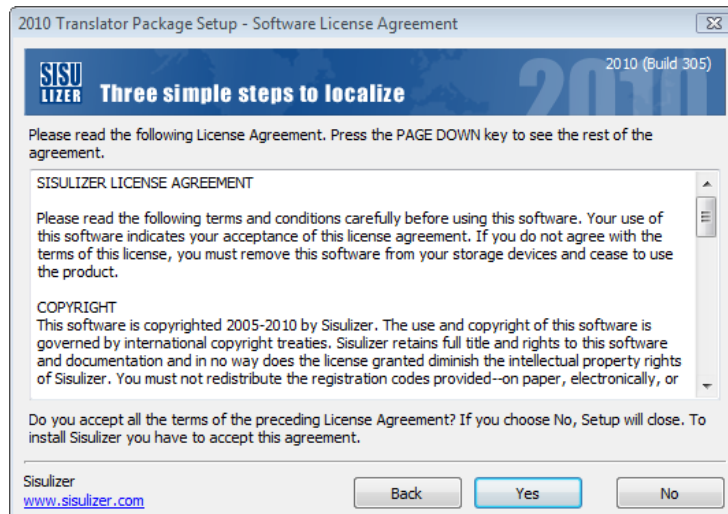
If it is a file ending with `.exe`, you can simply double-click on it. This will start an installer which installs a Free Edition of Sisulizer and opens the project. Follow the instructions of the installer and you will be fine:



This is the Free Edition installer. It will install everything you need to translate with Sisulizer in a few simple steps.

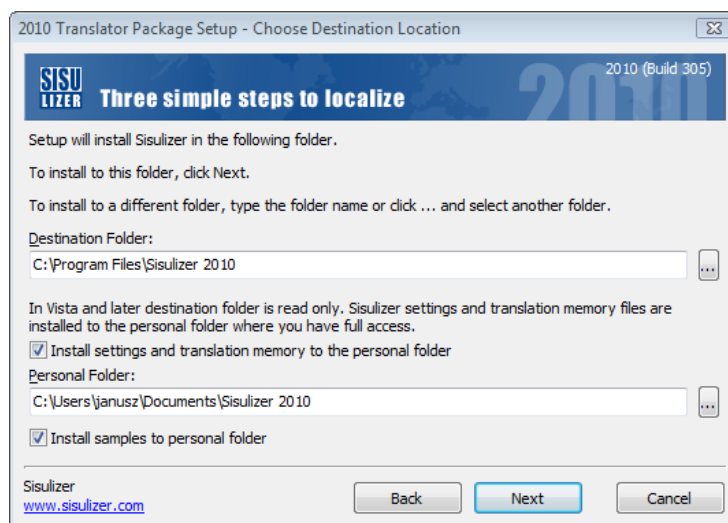
Just follow the instructions on screen and you should be fine.

Next, read the license agreement carefully. If you agree, click **Yes** to continue.

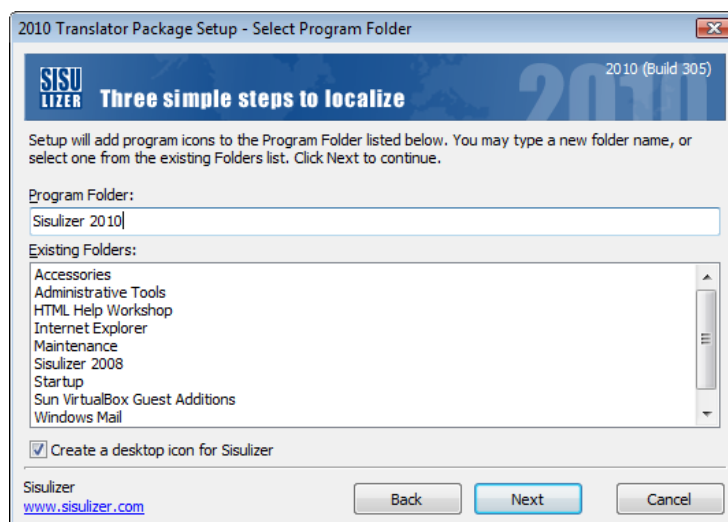


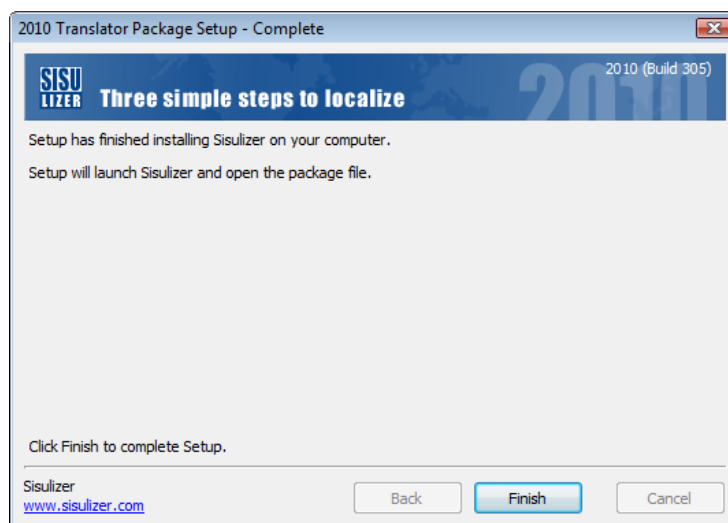
Now, you have to choose a path where to install the Free Edition. If you're not sure, leave the default and click **Next**.

If you install Sisulizer under Windows XP there will be no path selection for the sample files in the next step.




Next, select a start menu entry. You can start the Free Edition later by clicking in this start menu entry. Again, if unsure, leave the default and click **Next**.





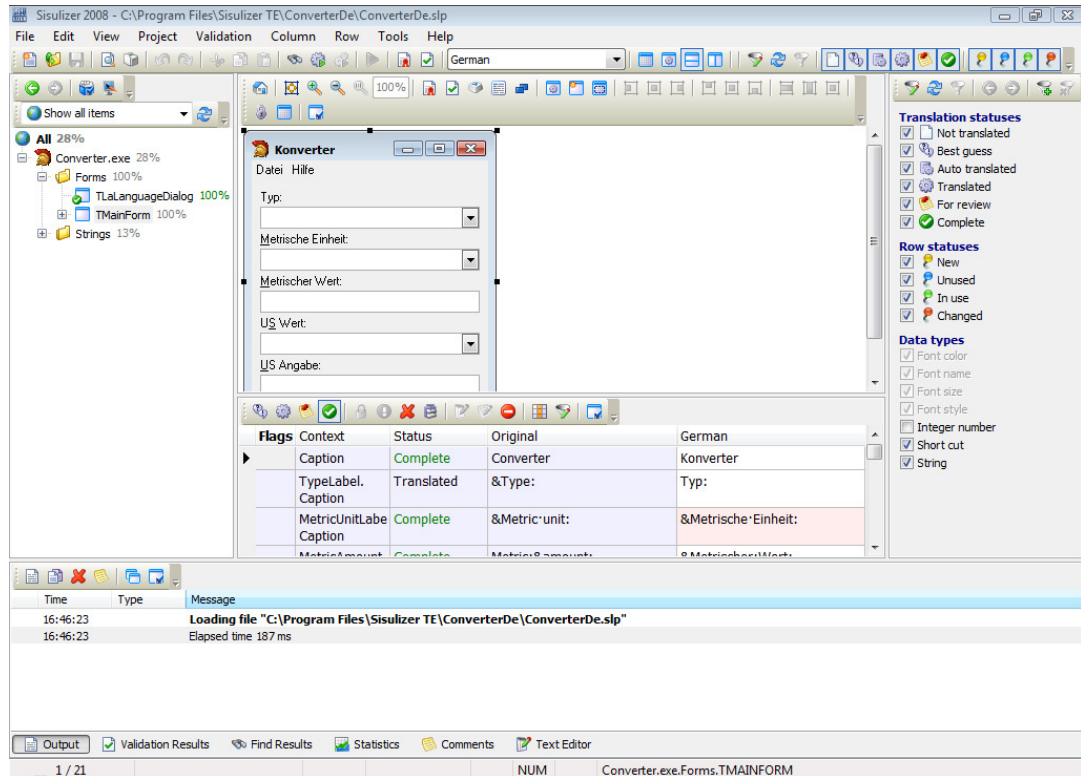
If you get errors, be sure to ask your administrator for the appropriate permissions.

If you've received a file ending with `.sle`, first have a look at the file's symbol. If it looks like this:  you've probably Sisulizer installed. Just double-click and Sisulizer will open the project. If it looks different, ask the sender about a Translation Version (it's free of charge).

Experienced users can also download a Free Edition from <http://www.sisulizer.com/downloads.shtml>. When asked for the version you want to install, select Free Edition.

The Free Edition has some limitation. For example validation is only available in the Developer Editions or the Translator Edition.

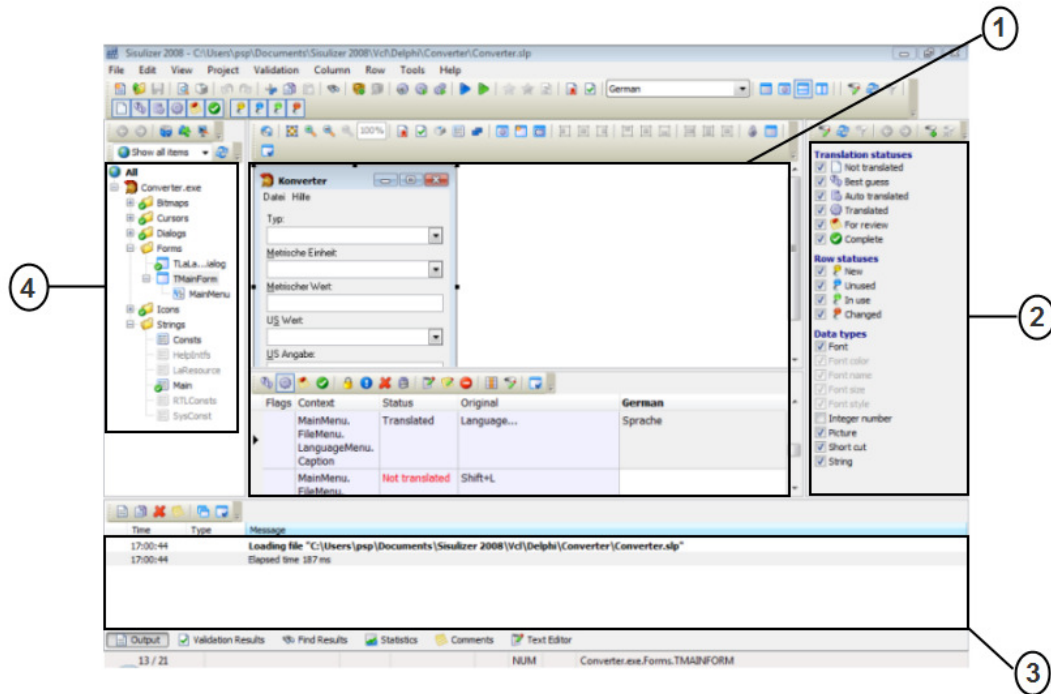
After installing, Sisulizer Free Edition automatically opens the project. What you see on screen should look similar to this:



Your main focus will be the big part in the center, called the translation grid. You can start translating right away; there are no more obstacles to take.

You will learn more about the translation grid in the next topic.

2.2. Start to translate (the translation grid)



After opening your Sisulizer project, you will see a user interface, which is divided into four parts: the translation sheet, the information pane and the project tree.

The four main parts are the following:

- ① The translation sheet and WYSIWYG display.

Here your translators do all their work. Sisulizer has WYSIWYG capabilities for all visual components like dialogs and menus.

- ② The filter pane.

You can quickly filter out certain view conditions using this pane. We will explain this in detail later.

- ③ The information pane.

You'll see error and success messages as well as statistics, comments and validation results.

- ④ The project tree.

As you have seen, software consists of several resources, like icons, strings, dialog boxes and so on. Here you see an ordered view of resource types available. It controls what you can see in the translation sheet. By default this will be everything translatable. Working on

a single node will keep you more aware of the context of the translation. Also, dialogs will only show up visually if you're working on the "dialog" node.

Everything is set up to begin translating now.

Nearly everything in Sisulizer has context menus. Use a right-click the elements with the mouse to see selected features related to it.

2.2.1. The translation sheet

Let's have a closer look at your main working place: the translation sheet:

Flags	Context	Status	Original	German
	Converter.exe. Forms. TLALANGUEDIGI OkButton.Caption	Not translated	OK	
	Converter.exe. Forms. TLALANGUEDIGI CancelButton. Caption	Auto translated	Cancel	Abbrechen
	Converter.exe. Forms. TMAINFORM. Caption	Not translated	Converter	
	Converter.exe. Forms. TMAINFORM. TypeLabel. Caption	Translated	&Type:	Typ:
	Converter.exe. Forms. TMAINFORM. MetricUnitLabel.	Not translated	&Metric'unit:	

This is a section of the user interface you see in the above screenshot (slightly modified).

It shows a grid with the last two columns being the original and translated strings. This is where you type in your translations.

Two rows are already translated into German language. You can see that from the mostly empty column to the right. This is the target language column.

Two entries are filled in here.

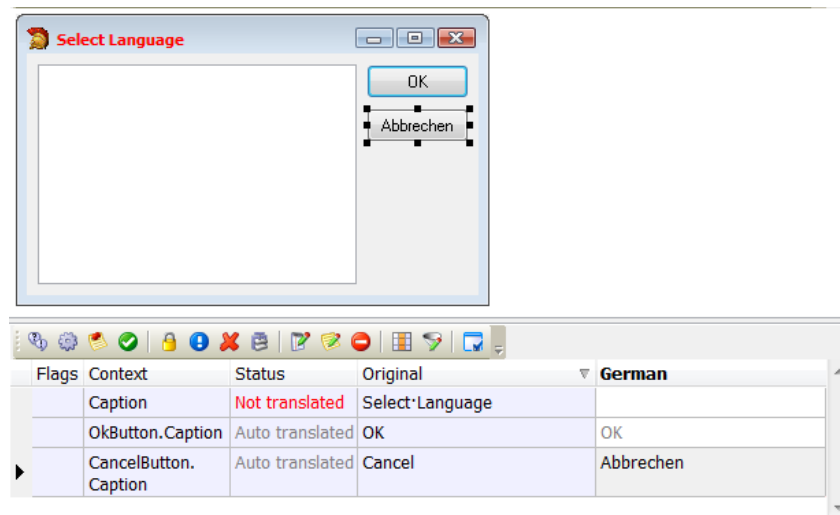
The first entry is automatically translated from another row which also says "Cancel". Sisulizer automatically translates duplicates and tags them as `Auto translated`. Check the grey entry in the `Status` column.

The translation has a light red background, which signals that the translation is longer than the original. The bigger the difference, the darker will be the red background. If it is a dialog, see if it still fits in the WYSIWYG display.

The second entry is manually translated and so is tagged as `Translated`.

2.2.2. WYSIWYG – visual translation

Let's have a look at how Sisulizer handles graphical dialog boxes:



This screenshot shows visual dialog box representation.

The dialog box occupies the upper part of the translation sheet. Red labels are not translated yet.

There are some labels already translated; have a look at the button row in the lower part of the dialog box.

Below the dialog box is the actual grid you know from the last screenshot.

Here, you will find the two translations of the above example again.

While translating, some strings may not fit in their containers, for example a translation for "Cancel" in some languages is wider than the button (the background of the translation will become very dark red).

This is not a problem with Sisulizer as you can even adjust the layout. Just click on an edge of the button and drag it until it's the size you need. Don't worry; the layout will only change in the localized version, not the original software.

2.2.3. Editing the translation

With this information you can start translating. Position your mouse pointer into an empty field, click and start typing. Hit Enter when you are ready.

You will notice the "Status" column in the current row will change from `Not translated` to `Translated`. You can also set the status manually. Read the next topic to learn more.

To jump right to the next empty item, press `<Ctrl+G>` which is holding the `<Ctrl>` key on your keyboard down and then press the `<G>` key. Sisulizer will automatically find the next empty item for you to translate.

Sometimes you will have to use the original as your translation. For example if you translate "OK" into German, you will translate it as "OK".

To copy the original, press `<Ctrl-Alt-C>`, which is holding the `<Ctrl>` and `<Alt>` keys simultaneously and then pressing the `<C>` key.

This and the above mentioned fact that duplicates are translated automatically makes it really fast to translate with Sisulizer. You will see, you'll be through a small application on no time.

Read on to get an overview about what the `Status` column means and what you can do with it.

2.3. Translation status

With Sisulizer you will always be aware of the context of your translation. In a perfect environment, translations can be either translated or not translated. But unfortunately, it's not always perfect.

Sisulizer deals with such situations by providing different translation states.

2.3.1. Not translated

If the project is untouched, everything is in state `not translated`. If you enter a translation, it will be in status `translated`.

You can always see the translation status in the `Status` column of the translation grid.

2.3.2. Auto translated

Now, what if you have a ton of dialog boxes to translate and they all have the typical buttons "OK" and "Cancel". Fortunately, Sisulizer takes notice of this and translates further occurrences of "OK" and "Cancel" automatically.

But you don't want these translations to be exactly the same as your manually translated ones. Maybe one "Cancel" occurs in another context and the translation is slightly different. Sisulizer marks these translations as `Auto translated`:

Font			
Dialogs.11.2. Text	Auto translated	Cancel	Abbrechen
Dialogs.11.2.	Not translated	True	

If you change the translation, the status will change to `translated`, so you can distinguish between these two easily.

2.3.3. Why translated is not the same as complete

If you are working on a small project, there is no extra quality assurance and you're 100% sure you translated everything correct, then you can treat `translated` as being `complete`.

That's not always the case. In general, the projects will be bigger, there will be quality assurance and after a hundred translations you may have made one or the other mistake.

Somebody will proofread your translations, be it yourself or another person. They also need a mechanism to mark translations as "check, I've done this". So, there is this project where everything is in `translated` status, proofreaders do their work and mark correct translations as `complete`.

If some translations need correction, they could change the status to `for review` and send it to the translator (which is you). You can quickly identify `for review` states, correct them and send them back. Read on to learn more about that.

So instead of looking through the whole project (remember, not small) and finding the needle in the haystack, you can see everything in an overview.

2.4. Getting an overview: Filter

To get an overview of translation statuses, use the easy to use filters:



This is the translation status panel on the right of part of Sisulizer.

It determines the translations to be displayed. Uncheck everything but `For review` to only see translations with this status.

The same goes for the proofreader. If you do a correction, it will be in `translated` Status. As his proofreading will result in `complete` status, he can filter for the `translated` and have a quick overview.

Flags	Context	Status	Original	German
	Forms. TLALANGUED Caption	Complete	Select*Language	Sprache*auswählen
	Forms. TLALANGUED OkButton. Caption	Complete	OK	OK
	Forms. TLALANGUED CancelButton. Caption	Complete	Cancel	Abbrechen
	Forms. TMAINFORM. Caption	Translated	Converter	Konverter
	Forms. TMAINFORM. TypeLabel. Caption	Translated	&Type:	Typ:
	Forms. TMAINFORM. MetricUnitLabel. Caption	Complete	&Metric*unit:	&Metrische*Einheit:

Translation statuses

- ☐ Not translated
- ☐ Best guess
- ☐ Auto translated
- ☒ Translated
- ☒ For review
- ☒ Complete

Row statuses

- ☒ New
- ☒ Unused
- ☒ In use
- ☒ Changed

Data types

- ☒ Font color
- ☒ Font name
- ☒ Font size
- ☒ Font style
- ☐ Integer number
- ☒ Short cut
- ☒ String

Special filters on!

This screenshot shows a translation grid filtered by `translated`, for review and `complete`. Here, the translations for "Converter" and "&Type" wait to be proofread and set to `complete` status.

You can change statuses with the status toolbar:



As you can see, there is a status, we didn't talk about yet: `Best guess`.

This comes into play if there are two possible translations which both fit the context. Mark it as `Best guess` to show you have chosen one translation but another one may also be correct.

Translation statuses also play a role when it comes to statistics. Statistics are good to check the progress of the localization. Read the next topic to learn more about statistics.

2.5. Comments

You can put comments to your translations. This is handy, if you are unsure about a certain translation. You can then put the other possibilities into comments. Proofreaders can then look up the comments and find an appropriate translation.



Click the comment button to add a comment to your translation.

Commented translations will have a small red triangle in the upper right corner:

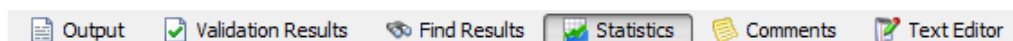


Please keep in mind that Sisulizer has additional possibilities to also add comments to rows and nodes. These are not described here.

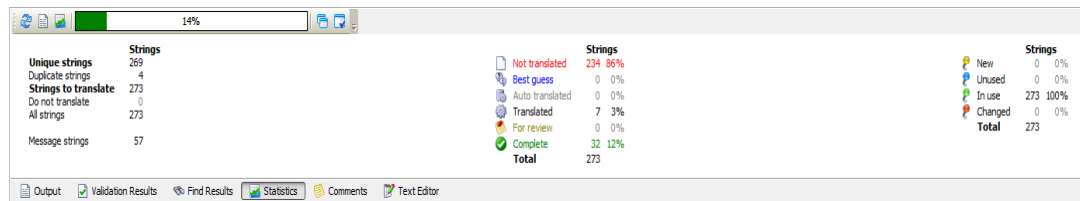
2.6. Check your progress (statistics and reports)

There are times when your boss wants to know how much of your translation is ready. Bosses like answers like: "I've already done 80%". With Sisulizer you can give him this information and have a proof that you can give him.

In the lower part of the Sisulizer user interface you'll find a panel which has five buttons:



These buttons each open a different display. Click on `Statistics` and Sisulizer will open the Statistics panel:



There is a colored bar on the top and numbers below. The colors of the bar correspond to translation statuses (statuses are covered in the previous topic). You can see the color code below.

It's best to use the colored bar only for a quick overview. Here, it's mostly red, which means most of the project is still not translated. Below the bar you can see the numbers you want:

There are 79 strings not translated, which is 74% of the project. So, you can say you've done 26% of the project. But you can be more distinctive.

You can also see that 19% is translated (which most of the time means manually translated - see the previous topic for details). 3% is auto translated, these are for example duplicates. Sisulizer translates duplicates automatically and assigns the status `auto translated` to them. Sisulizer also displays the amount of duplicate strings.

Another 5% of the project is `complete`, which usually means there was proofreading involved.

So, in this example you can tell your boss "I've done 26% and 5% is already proofread."


Be aware that the statistics view depends on three factors:

- The language you have selected.
- The node you have selected in the project tree.
- The filters you have applied.

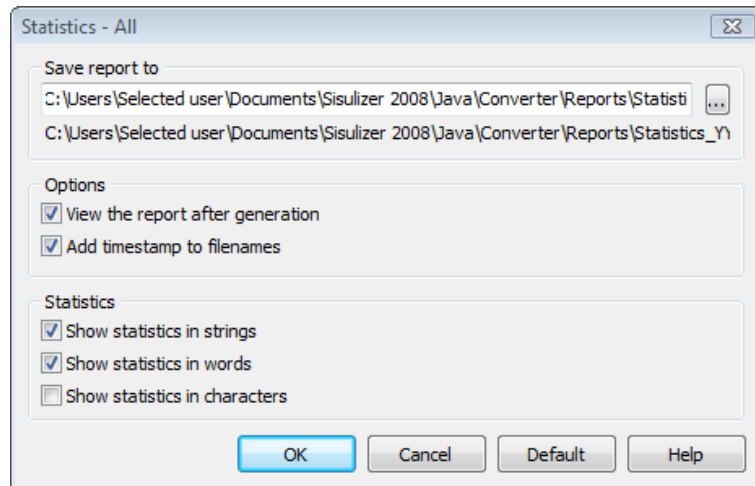
2.7. Reports

To give other people a detailed view of your progress, you can create HTML reports. These reports consist of a single file and will open in a browser.

In contrast to statistics, reports do not depend on the filters or node you have selected. They are always complete overviews of your progress.

To create a report, click on the report button: 

A dialog box shows up which asks you for a filename and some options:



Enter a filename to save the report to. It is possible to add a timestamp to the name.

You can customize the report by using the four options:

- Show statistics in strings: This will tell you how many strings you have translated. Strings can be single words like "Cancel" or a bunch of words like titles in dialog boxes. "Project Statistics Report" is a string.
- Show statistics in words: This will tell you how many words you have translated. This is definitely the most interesting information when it comes to software localization. The title of the dialog box from above would be segmented into the words "Project", "Statistics" and "Report".
- Show statistics in characters:
- If you ever want to know how many characters you have translated, you can be sure Sisulizer gives you the answer.

View the report after generation:

This opens the report in your browser when Sisulizer has generated it.

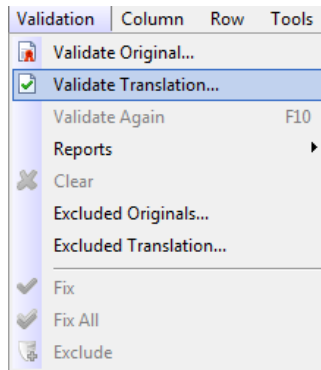
Statistics are an important issue if you are going to bill your work.

If your statistics tell you that you have 100% of your work done, read on to learn about what to do with your work. You will learn how to validate your translations and send the project back to the person you've received it from.

2.8. Automatic validation

Sisulizer can detect minor flaws in translation – for example a missing keyboard shortcut (like <Ctrl+G>) - and will also automatically correct them if you say so.

Let's take a closer look at how validation works:

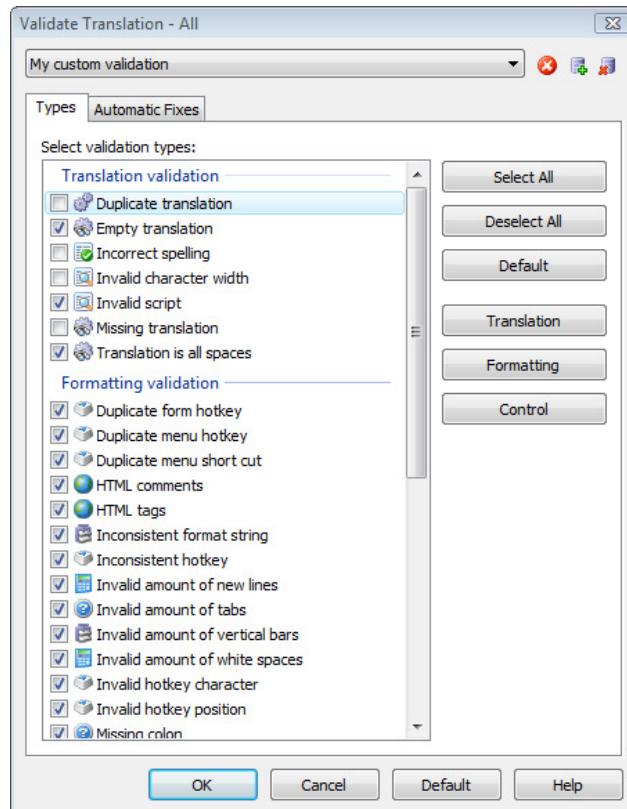


Click on `Validate Translation...` in `Validation` menu to start the validation.

If you start validation for the first time, Sisulizer lets you set the validation options.

To use these options again, click on `Validate Again` in `Validation` menu, one entry below `Validate` in the screenshot.

Sisulizer will open a dialog box with which you can set what to validate:

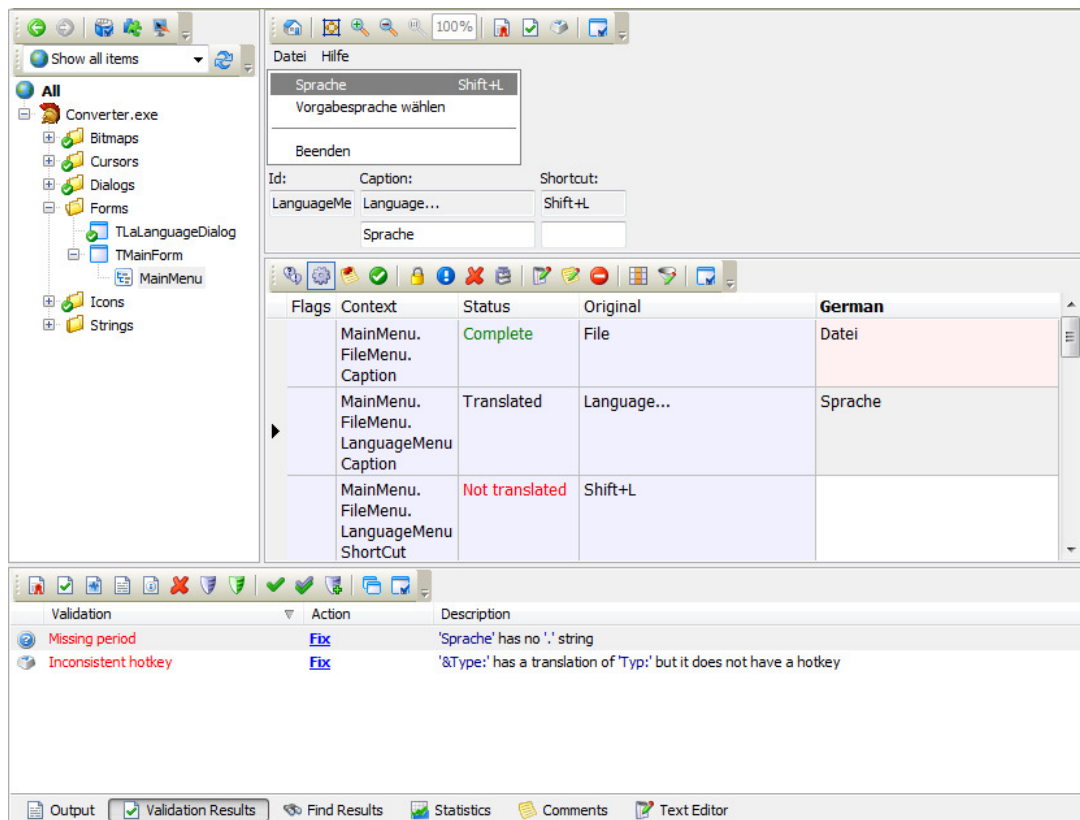


The default set is quite reasonable. It will check, amongst other things, for missing and invalid hotkeys.

If you need a detailed explanation of the options, simply press <F1> key to open the online help in Sisulizer.

Click **OK** to start the validation.

In our example case we have two minor translation issues. The issues will appear in the information pane and Sisulizer jumps right to the location of the issue:




In the screenshot above you see the two issues in the bottom part of the screen. These are:

Missing periods: User interface designers mark menu entries with three periods, if they open a dialog box. The translation does not have the three periods.

Inconsistent hotkey: The original is a menu entry and has a hotkey assigned. In this case, if the user opens the menu and presses <Ctrl+T> on his keyboard, this menu entry will be activated. The translation is missing such a hotkey.

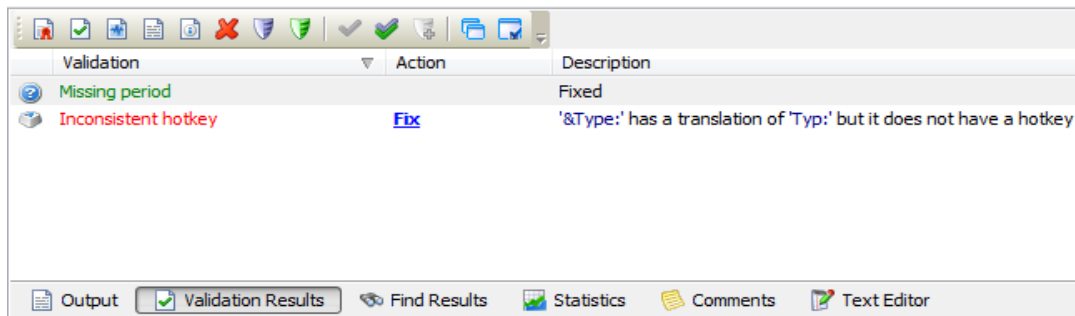
Sisulizer automatically selects the first of the issues and shows the location of this issue. As you see, it shows the menu with the menu entry selected we have to take care for.

These two example issues can be handled automatically.

To let Sisulizer correct the selected issue, click on the button: . Sisulizer will automatically add three periods to the translation:

	MainMenu. FileMenu. LanguageMenu. Caption	Translated	Language...	Sprache...
--	--	------------	-------------	------------

The information pane will show you this issue as `fixed`:

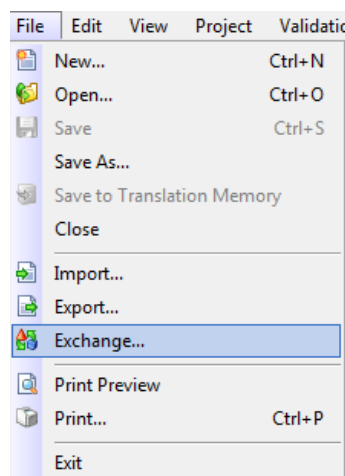


You can correct all issues found by clicking on: .

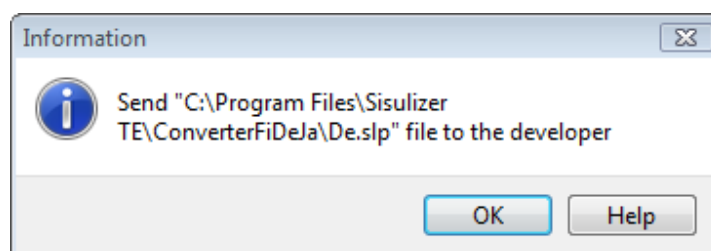
2.9. Finish your work

So, your translation is done, the statistics tell you 100% ready (or 0% not translated), you have successfully validated and you want to deliver the work to your client.

It's very simple. You only have to open the file menu and click on `Exchange` . . .



A message box will tell you which file to send back to the developer:



Put the file onto some CD or send it by email to the person who sent you the translation package.

That's all. You can have lunch break now.



3. Build

If you use the Free Edition you can skip this chapter.

This section tells you how to gather all the translations and import them into your Sisulizer project. Once done, you're ready to create the localized versions of your software with Sisulizer. This part of the process is usually done by the same people that do the "Scan" part.

You will learn about the following topics:

- Put the pieces together (merge translations into your project) (page 50)
You have gathered all the parts you created in the "Scan" section. Now you're ready to import them into your project. This topic will show how you can do this.
- Check the progress (statistics and reports) (page 54)
Before building your localized application you should make sure that all the translation is done. Get a quick overview of the translation progress by interpreting statistics. Get a more detailed summary by creating reports.

- Automatic validation (page 55)

You will also learn how to validate the translations and have minor issues fixed automatically.

- Selecting the output format (page 60)

There are four different ways you can build a localized application: a single localized file for each language, one big multilingual file which contains all the languages, one executable and a resource DLL for each language. Finally you can have the latter embedded into your executable.

You can combine these four output formats as you like and this topic shows you how to do it.

- Create the localized version of your software (page 61)

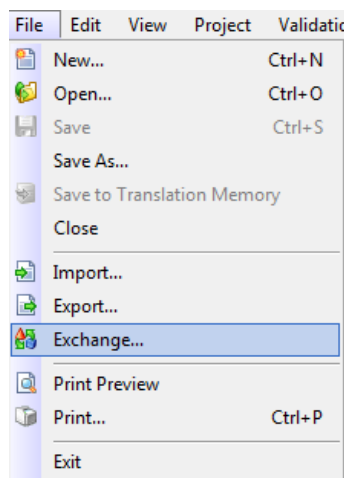
Finally you are ready to build your localized application. In fact, you're just one click away from a fully translated version of your software. Read this topic to learn how you can do it with Sisulizer.

3.1. Put the pieces together (merge translations into your project)

You have created a Sisulizer project and sent it to - maybe several - translators. Now you receive something from them which, they say, is the final translation.

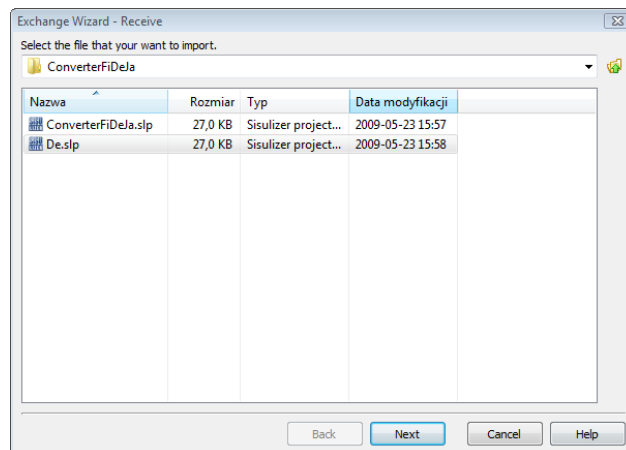
This is not a Sisulizer project but their share of the work. It's an exchange package with completed translations and you will have to merge it into your existing project.

Open your project in Sisulizer.



Once you opened it, click on `Exchange . . .` in the `File` menu.

This will open `Exchange Wizard`. There are five options to choose from. Click on `Import translated project file` and hit `Next` to continue.



Browse to the location where you've saved the exchange package you got from your translator.

Select it and click on `Next`.

This will open a dialog box with import options. The default options will be what you want in most situations. More information is available in the online help.

Nevertheless, we will have a closer look at the most interesting options:

In the upper part you'll find the import method. There are three options available in the drop down list:

- **Import by context:**
Sisulizer uses a context to identify each resource item. For example if you have a dialog resource which is called `TMainForm`, then its caption will have the context `TMainForm.caption`.

Because the exchange package is directly derived from your project, both will have the same context values.

When Sisulizer imports the exchange package, it will match the context of each entry in the package with the context of each entry in the project. Sisulizer will not import a translation until it finds a matching context.

- **Import by value:**
Sisulizer projects (and exchange packages) don't only save the context but also the value of the original language.

When Sisulizer imports the exchange package, it will look for an original language value, for example "Cancel". Then it will look into the exchange package and import a translation matching "Cancel".

This method does not check any context matches!

- And what does the third option do?

It imports first by context, meaning that it matches contexts like mentioned above. Then, if it does not find matching contexts, it will import by value. This will probably never happen when importing exchange packages from translators. But it may be useful if these packages come from other projects.

Option "Import also those items where the translated value equals to the original value":

☒ Import also those items where the translated value equals to the original value

When you import translations where the translation is the same as the original, this will import the translation nonetheless.

For example, in some countries the "OK" button translates to "OK". Importing without this option would result in an empty translation with status "not translated". This is probably not what you want, so keep the option checked.

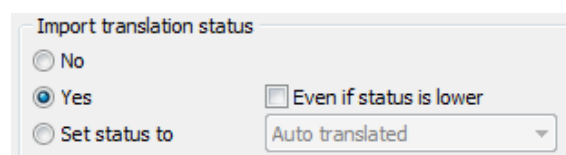
Option "Import also those items that are empty":

☐ Import also those items that are empty :

If checked, Sisulizer imports empty translations and will probably overwrite existing translations.

Imagine you've sent the translators an empty project and did some translations yourself afterwards. Maybe you translated all buttons "Cancel" (which can be done very quickly). You informed your translators about that and they left these buttons out.

If the above option is checked, Sisulizer will import their translation, which is empty. The result will be no translation.



Be sure to have selected `Yes` here. If your translators have done everything right, translations will be at least of status `Translated`.

If you select `No`, the translations are imported but the status is not updated. You could end up with translations being present but in status `Not translated`.

If you are really sure, you can set the translation status to a fixed value with the third option. In general, don't do that.

Click `Finish` to start importing the package.

Now you can have a look at the statistics if you want to be really sure that everything is translated. Read more about that in the next topic.

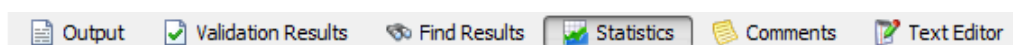
If you have more than one package, repeat the steps mentioned here with every package. Usually it makes sense to have one package for each language.

3.2. Check the progress (statistics and reports)

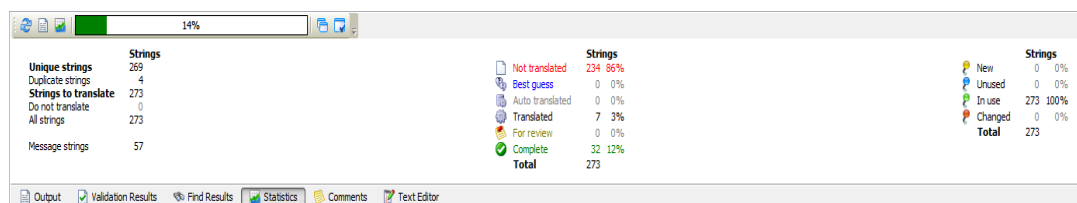
Before you create your localized software, you should have a quick look at the statistics to be sure everything is complete. Of course, it's good to trust your translators but for example they could have sent you an older backup of the project by mistake.

It's best you check the statistics after importing each exchange package. If there is something missing, you will know who to contact.

In the lower part of the Sisulizer user interface you'll find a panel which has five buttons:



Click on `Statistics` and Sisulizer will open the Statistics panel:



There is a colored bar on the top and numbers below. The colors of the bar correspond to translation statuses (statuses are covered in the previous topic). You can see the color code below.

It's best to use the colored bar only for a quick overview. Here, it's mostly red, which means most of the project is still not translated. Below the bar you can see the numbers you want:

There are 79 strings not translated, which is 74% of the project. So, you can say you've done 26% of the project. But you can be more distinctive.

You can also see that 19% is translated and 3% is auto translated, these are for example duplicates. Sisulizer translates duplicates automatically and assigns the status "auto translated" to them.

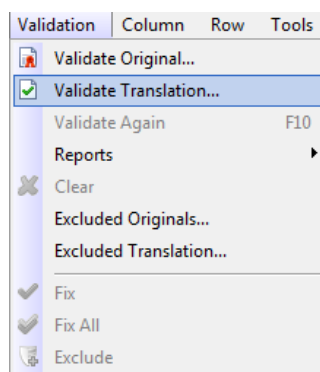
If you import exchange packages one at a time (as we suggested earlier), you may have such a high number of "not translated" strings, although the part you've just imported is complete. This is because the parts, which you did not import yet, also count in the statistics. The statistics view depends on three factors:

- The language you have selected
- The node you have selected in the resource tree.
- The filters you have applied.

3.3. Automatic validation

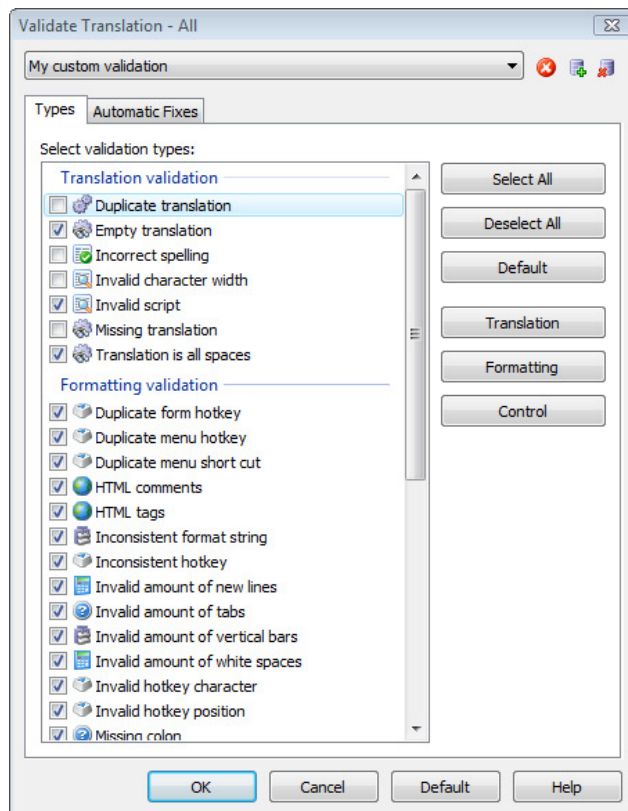
Sisulizer can detect minor flaws in translation – for example a missing keyboard shortcut (like <Ctrl+T>) - and will also automatically correct them if you say so.

Let's take a closer look at how validation works:



Click on `Validate Translation...` in `Validation` menu to start the validation.

Sisulizer will open a dialog box with which you can set what to validate:

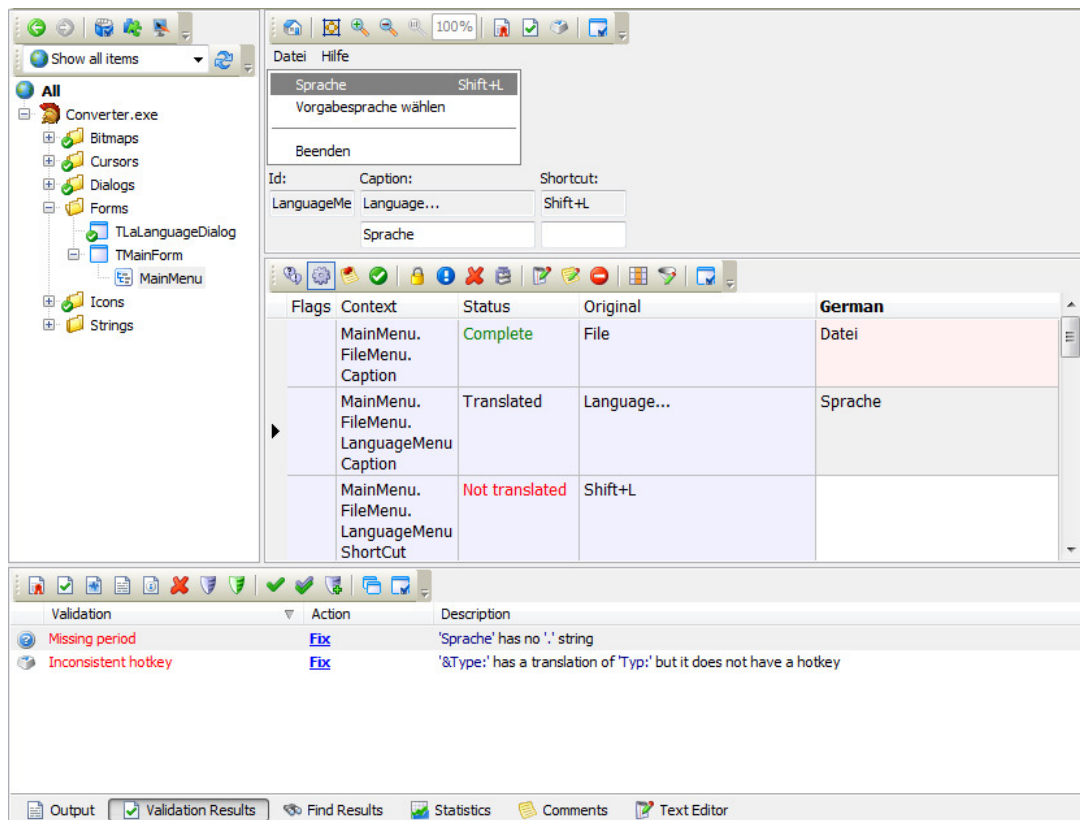


The default set is quite reasonable. It will check, amongst other things, for missing and invalid hotkeys.

If you need a detailed explanation of the options, simply press <F1> key to open the online help in Sisulizer.

Click **OK** to start the validation.

In our example case we have two minor translation issues. The issues will appear in the information pane and Sisulizer jumps right to the location of the issue:




In the screenshot above you see the two issues in the bottom part of the screen. These are:

Missing periods: User interface designers mark menu entries with three periods, if they open a dialog box. The translation does not have the three periods.

Inconsistent hotkey: The original is a menu entry and has a hotkey assigned. In this case, if the user opens the menu and presses <Ctrl+T> on his keyboard, this menu entry will be activated. The translation is missing such a hotkey.

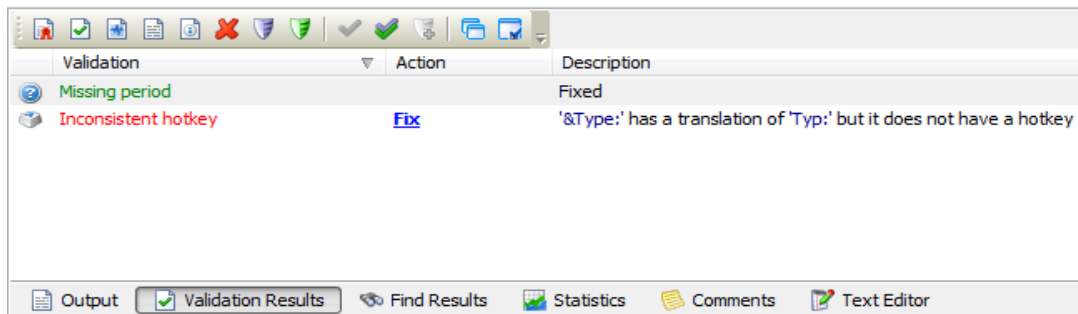
Sisulizer automatically selects the first of the issues and shows the location of this issue. As you see, it shows the menu with the menu entry selected we have to take care for.


These two example issues can be handled automatically.

To let Sisulizer correct the selected issue, click on the button . Sisulizer will automatically add three periods to the translation:

	MainMenu. FileMenu. LanguageMenu. Caption	Translated	Language...	Sprache...
--	--	------------	-------------	------------

The information pane will show you this issue as *fixed*:



You can correct all issues found by clicking on .

Your translator should have already done the validation if he uses the Translator Edition and also should have fixed problems he found. But to be sure, always validate yourself before building the localized version if your translator uses the Free Edition. Choose the right output format

Sisulizer supports four different output formats for localized applications:

	Results in multiple files	Results in a single file
Automatic language selection	<p>Single localized files</p> <p>This is the default format. Sisulizer will create a separate executable for each target language. Each executable will be put into a different directory.</p>	<p>One multilingual file</p> <p>Sisulizer will create one large executable containing every language you translated. The executable will be put into a specified directory.</p>
User language selection	<p>One executable, several resource DLLs</p> <p>Sisulizer will create a separate executable and a resource DLL for each language.</p> <p>Be aware that the executable is actually the same for all languages. The files will be put into a different directory each.</p>	<p>One executable with embedded resource DLLs</p> <p>Sisulizer will create resource DLLs internally and embed them into one application file.</p> <p>The first execution of the localized application will extract the embedded resource DLLs.</p>

Which one is the best?

Of course, there's no definitive answer. You can choose to have one big file with all translations or several small files for each language. The first row of the table represents this decision.

Second, you'll have to decide if your users should be able to change languages at runtime. The bottom row of the table represents your decision.

You should also consider the following:

- **Size:** This is important to optimize download sizes if you'd like to distribute your software over the internet. Although they share similar content, single files are typically smaller than two or more files.

In contrast, you can optimize by letting your customers download a core version and an additional language DLL which makes the overall size smaller than downloading all languages at once.

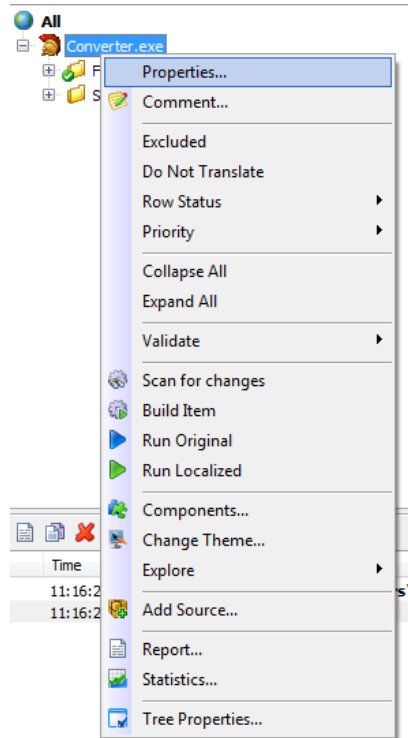
- **Versioning:** New versions of software may be substantially different from former versions. This leads to certain incompatibilities between DLLs and the application if the versions don't match.

Usually, distributing a single file is a good solution to this issue.

You can combine these four output formats as you wish. If you cannot decide now, you could simply generate all of them. They all appear in different subdirectories which are again fully customizable.

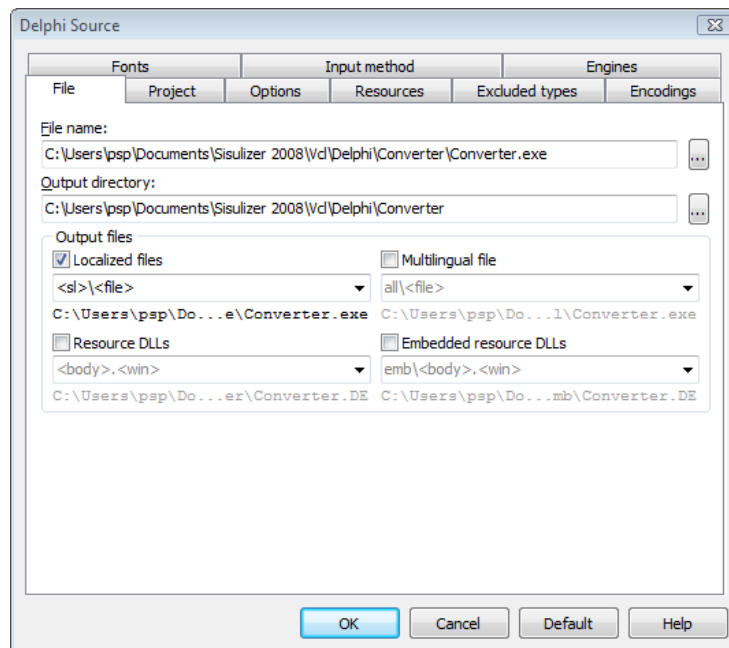
3.4. Selecting the output format

The output format is an option in the properties of your project. Here is how you change them:



To make your decision, right click on the executable name in the project tree and click on Properties...

This will open the source options dialog box. You will see the above table represented in the center part of the dialog box:



The output paths are customizable. There's a good explanation on format strings in the online help.

3.5. Create the localized version of your software

You have everything translated and set your favorite output format. Now you are ready to create your fully localized application.

There is no need to recompile your sources or even open your development environment. You can do it all in Sisulizer.

Have a look at the toolbar and find the following part:



What you see here is:

- Build item: Creates your localized application according to your chosen output format.
- Run original source: Starts the original application.
- Run translated source: Creates you localized application as above and run it afterwards.

That's it! Now you only have to distribute your localized application and conquer foreign markets.