



MozillaTranslator v5.30

Table of Contents

User Guide.....	4
Introduction.....	4
Installation.....	5
First run.....	5
Products, directories and JAR.....	6
Updating the product.....	6
Editing strings.....	7
Editing phrases and images.....	8
Searches, auto translation, and queries, a lot of queries.....	9
Labels, commandkeys and accesskeys.....	10
Exporting JAR, XPI and SCM.....	11
Migrating a product.....	12
Working in group.....	12
Menu Reference Guide.....	14
File Menu.....	14
Manage Products.....	14
SCM info panel.....	14
JAR info panel.....	15
Update Product.....	15
Save Glossary.....	16
Save Settings.....	16
Setup.....	16
Display Tab.....	17
Input/Output Tab.....	17
Translation Assistance.....	18
Key Connections Tab.....	18
Automated tests Tab.....	19
Quit.....	20
Import menu.....	21
Import Partial Glossary.....	21
Import File.....	21
Import Translation from JAR.....	22
Import Translation from SCM.....	22
Export menu.....	23
Export Partial Glossary.....	23
Export File.....	23
Create XPI File.....	23
Localizing the XPI installation messages.....	24
Create JAR File.....	24

Export in XML Format.....	25
...File.....	25
...Component.....	25
...Glossary.....	25
Export SCM-based Translation.....	25
Edit Menu.....	26
Chrome View.....	26
The Show What dialog.....	26
The Chrome View Window.....	27
The tool bar.....	28
The tree.....	30
The Content tab in non-image files.....	30
The Content tab in image files.....	31
The Information tab.....	32
The new Auto-assign Accesskey Dialog.....	32
Auto-assign Acceskeys Preferences Dialog.....	33
How it works.....	34
Some use cases.....	34
Automated Test Queries.....	35
Untranslated Strings.....	35
Check Keep Original with trns.....	35
View Original == Translated.....	35
Check Variables.....	36
Check Key binding.....	36
View DONT_TRANSLATE not honored.....	36
Check for different endings.....	37
Check for duplicated words.....	37
All above at once.....	37
Check for mismatched pairs.....	37
Redundant Strings.....	37
Search Translation Inconsistency.....	37
Search similar strings (experimental).....	38
Relink Labels & Keys.....	38
Clear Fuzzy Flag.....	39
View Fuzzy.....	39
View Binary.....	39
Search.....	39
Advanced Search.....	39
Replace.....	40
Edit Selected Phrase.....	40
Basic Tab.....	40
Advanced Tab.....	42
QA results Tab.....	42
Auto-Translate.....	43
Advanced Menu.....	44
Migrate Product.....	44
Assign License.....	44
Help Menu.....	45

About MozillaTranslator.....45
Keyboard shortcut reference.....46
Main menu options accelerator keys.....46
Common options shortcuts.....46
Special combinations.....47

User Guide

Introduction

MozillaTranslator is a Java application designed for helping localizers in maintaining and leverage their work with Mozilla based applications.

The quickest overview about how to use MozillaTranslator to localize a Mozilla-based product from scratch can be summarized as:

1. Get local copies of en-US and your locale (eg. Ab-CD) repositories.
2. Download and install MozillaTranslator.
3. Go through **File | Setup** to customize the program behavior to your taste. If you are not sure what something means, leave the defaults, they should be a sensible choice in most cases.
4. Define one or more products you intend to translate. This is little more than give a product name and some paths to the localizable files.
5. Run **File | Update Product** to fill in the product strings to localize, or to update them, adding, modifying and removing MozillaTranslator's database.
6. Translate strings using some of the different views (Chrome View is fine to start a new localization).
7. Save your work (**File | Save Glossary**) as you need it (MozillaTranslator works with in-memory data structures), stop, give yourself a rest and continue your work later.
8. Once it is finished, run automated tests (**Edit | Automated Test Queries | All above at once**) to let MT find for you some potential errors.
9. Export the translation using **Export | Export SCM-based translation**, or any of the other options as needed.
10. Test your localization by creating, installing and activating a language pack.
11. Fix the typos and errors you find during testing and repeat steps from 7.

The next pages go with more details for these points. In case you need even more detailed information, there is a Menu Reference Guide after this User Guide.

Installation

Follow these steps:

1. Get the most recent ZIP at MozillaTranslator page at Kenai.com: (<http://kenai.com/projects/moztrans/downloads>).
2. Unzip it in an empty directory (for example, "MT").
3. Make sure you have a working Java Runtime Environment 1.6 or better (you can get Oracle's JRE at <http://www.java.com/>).
4. To run MozillaTranslator on Windows, just double click on the JAR file. On other graphic environments, you could do the same.
5. If the above doesn't work in your system, you can run it from the console shell (either Windows or Unix): `java -jar mt5xx.jar`, where `mt5xx.jar` represents the exact name of the JAR file.

Note: unfortunately, there is a non-traced memory leak that can lead to `OutOfMemoryException` crashes if working with moderate to big glossaries. To minimize and, in practice, remove the risk of running out of memory, you'll need to use java command line options `-Xms` and `-Xmx`, for example:

```
java -Xms64m -Xmx256m -jar mt5xx.jar
```

MozillaTranslator (from herein, "MT") looks for their data and preferences files in the current working directory, so if you want to keep apart some products or modules from others, you may run MT with commands like: `java -jar ../../mt5xx.jar` and so.

Products based on JAR sources may want to use a JAR directory to put exported JARs, and a XPI directory for exported XPIs.

First run

After the main window opens, you should pay a visit to the **File | Setup** menu option. Here you can set MT preferences regarding display, input/output, Translation Assistance, Key Connections and Automated Tests. For detailed information, see the Reference Guide after this User Guide.

The preferences are saved in a file named `mozillatranslator.properties`. Your work in localizations will be saved in a file named `Glossary.zip`, which contains some directories for images (if there is any present in the translation), and one or more text files file with info from any other file (DTD, Properties, (X)HTML, etc.) of each and every product defined.

Products, directories and JAR

MT works with Products, and you need to create at least one product. You can think a product represents the group of localizable files used by an application. For each product you can collect the files archived in compressed, specially crafted JAR files (as *special* as Mozilla products require and ship with, nothing else; e.g.: en-US.jar), or in a directory and all of its subdirectories (which is used with a SCM repository, like CVS, SVN or Mercurial, hence the “SCM-based” product type).

You may have just one directory (and, thus, just one Glossary.zip), with all products you are going to translate, or you may have different directories for each product, with all versions in the same Glossary.zip. Or you may even separate both products and versions on their own Glossary.zip, as the proposal shown in Installation section.

Updating the product

Right after creating a new product, and every time you want to update it (because of new product versions in JAR-based products or recent changes to SCM-based products), you will have to run **File | Update Product**. You select one or more products and, if you select just one and it is SCM based, enter the import path (if you have set a default SCM Import path in the Product Properties, it will be automatically proposed for you). After accepting the dialog, you will be presented with the Select Which Columns dialog, and will get an Updated Strings list (actually, it has just new and modified strings; deleted strings and entire files will not be shown, but silently dropped from the Glossary.zip). The previous original value for modified strings will be shown in the QA results tab.

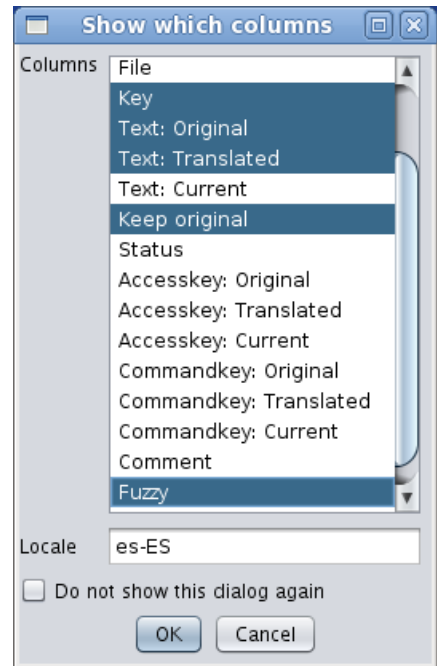
Note: if you are managing a SCM-based product, you need to pay attention to the SCM diff logs when comparing your local copy and the SCM repository: MT will NOT delete removed files automatically, so you will have to do it by yourself, as well as remember to export files with removed strings.

Every string new or modified will be marked as “Fuzzy”. Fuzzy strings are those that need review by translator; this status is saved in the Glossary.zip, so you can start an update and quit MT if you have not enough time to finish the review in a single session. You can run MT again later and go to **Edit | View Fuzzy** query to get all remaining Fuzzy strings.

Editing strings

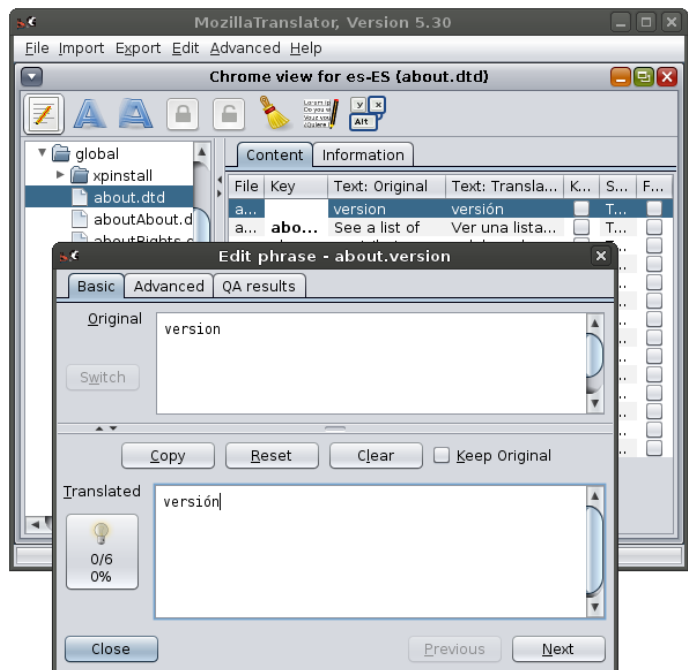
Whether you are editing strings in a result list, like “View Fuzzy”, “Untranslated strings” or any other query, or browsing the products through Chrome View, you will see most data in the form of a table with several columns. Before accessing the view, you will be queried about which columns to show (unless you have opted for hiding the dialog, that can be restored in the Display tab of Setup dialog).

Which columns to show will vary depending of the query run and your personal taste, but you will surely want to have always visible the key/entity name, the original value, and the translated value. Keep Original and Fuzzy flags are likely to be included most times, too. Other values may or may not be useful, depending of circumstances. To select several columns, keep [Ctrl] pressed while clicking the desired columns. Columns are described in the Reference Guide.



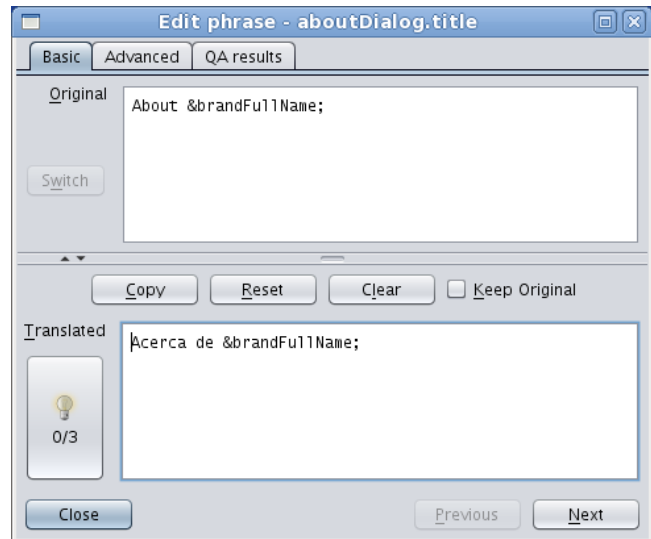
Note: besides selecting columns, this dialog allows you to select the locale. You positively **MUST** write the right locale code; writing it with a typo will result in empty translations without you knowing the reason, and panic will obfuscate your mind. :-)

Once you close the columns selector, you will see the Chrome View. The image to the right shows MozillaTranslator main window with Chrome View and the Edit Phrase dialog overlaid. The look and feel is the new Nimbus, available since Java 6 Update 10.



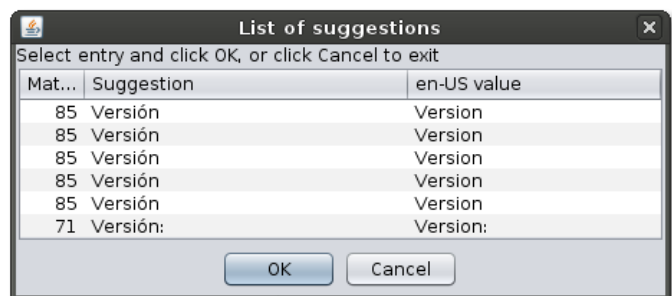
Editing phrases and images

The table and Chrome views are useful in many scenarios, but are somehow limiting when the original string is too large to fit in the window, or when it includes several lines. However, you may activate an advanced view in the shape of a separate dialog, the so called “Edit Phrase dialog”.



The Edit Phrase dialog has some features unavailable in the regular table view. For instance, if a string has a Localization Note (a note in the original en-US file with info for localizers), you only will be able to see it in this dialog, by pressing the Switch button in the Original text section (disabled in the image, because the entity has no localization note).

The most prominent difference, though, is the ability to get translation suggestions. If the original string you're reviewing is equal or similar to one previously existing, the light bulb will appear on the left and you will be able to either click the button to get a translations list or just type [Alt]+[S] to cycle through suggestions. The image shows the list dialog, with the original and the translation values, and the coincidence match between the original value in that row and the string you are translating right now.



The image shows the list dialog, with the original and the translation values, and the coincidence match between the original value in that row and the string you are translating right now.

As a result of having more space for each string than in a single table row, Edit Phrase dialog is more useful for long strings, for reviewing comments generated by automated QA results, and for getting detailed info through the Advanced tab. In fact, if you are a keyboard fan, you may find the Edit Phrase dialog as convenient as the table view to move through the list, thanks to the available keyboard shortcuts.

Note: the Edit Phrase dialog automatically sets off Fuzzy flag in every visited string (the only field not visible in this dialog). Should you need to restore this flag for a string after reviewing it in the Edit Phrase dialog, just return to the table view and either click on the Fuzzy flag or, if

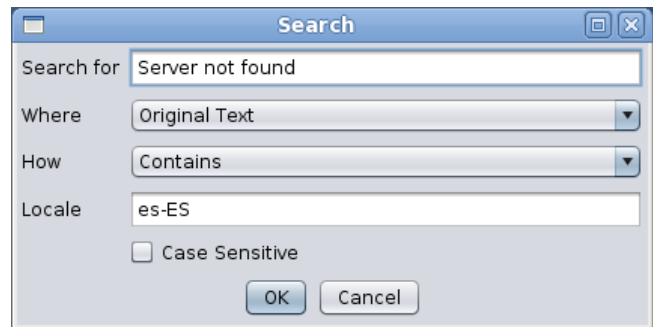
that column is not being shown, select the row and click the Fuzzy button.

Another fancy feature introduced with MT 5.20 is the embedding of the Image Panel in the Chrome View. The Image Panel is used to see image files present in localizations (as part as built-in help, for instance), and to provide localized versions of those images that you may have already prepared with a graphic editor (MT doesn't feature an image editor, sorry). The image to the right shows the standalone image editor but, if you are browsing an image file in Chrome View, you will get the same content just where a table is usually shown.



Searches, auto translation, and queries, a lot of queries

MT has a couple of search dialogs and a replace dialog. Both search dialogs allow to search for key names, original text, translated texts and user comments, and both allow to look for exact matches, partial matches and negations of those. The normal search dialog also allows to treat the search expressions as regular expressions; on the other hand, the advanced search dialog allows to specify up to three different search criteria that can be added (AND) or alternated (OR).



The Replace dialog works in a different way. You write what are you looking for **only in the translated strings**, what will be the replacement, and some other details (Exact Match means that the search string must be the whole content of a translation to replace it, it won't do it if the search string appears inside a bigger string).



There are also a bunch of queries. Most of them are used for Quality Assurance (QA) purposes; all can be accessed from the Edit menu, and all work pretty much in the same way: invoke the query, select which columns to be shown, and a window will open, which can be empty if no strings match the query.

The way MT understands a translation is that a string correctly translated is in one of these two cases:

- The original string needs a translation and it has been provided. Keep Original is not set, and the translated text is not empty.
- The original string is the same in en-US than in ab-CD. Keep Original is set and the translated text is empty.

Everything else will be detected in one of the QA queries as some kind of deviation (or, at least, warning).

Labels, commandkeys and accesskeys

As most moderns graphical programs, Mozilla applications allow access to most of its options by both mouse and keyboard. Accessing through the keyboard is done in two ways:

- keyboard shortcuts available through the whole program, or most of it. Usually they involve pressing [Ctrl] (or the equivalent Macintosh key) plus another key, and they are unique (the same combination always behaves in the same way inside the program). This kind of shortcuts are called “commandkeys”.
- Keyboard quick access keys available in menus and dialogs. They are represented as an underlined character of the UI label associated to the function. Sometimes they need to be invoked using [Alt] plus the underlined character. The same character is used in different dialogs, and sometimes even it appears several times in the same UI (although it is something to be avoided). These are the so called “accesskeys”.

The comandkeys don't need to have their character in common with the function they perform. For instance, you use [Ctrl]+[X] to cut text to the clipboard, and yet the menu option is **Edit | Cut** and thus doesn't have an **X** in their label.

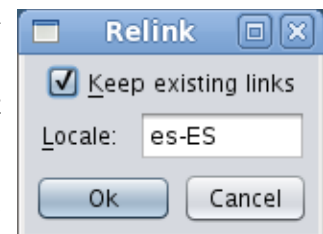
However, that's not the case with accesskeys. The **File | Open...** accesskey is O, but when you translate it into (for instance) Spanish, you have “Abrir...”, so if you let O being the accesskey, the menu option will be shown in the Mozilla UI as “Abrir... (O)”.

All of this means that there must exist a connection between UI labels and (at least) accesskeys, and that you need to care to adapt accesskeys to the label translation.

There are rules managing how to name strings that act as labels, accesskeys and commandkeys, mainly by adding suffixes for every of them. And, as every rule, they are sometimes not followed. :-) So MozillaTranslator 5.1 introduced some user preferences to let the localizer set a number of possible suffixes for labels (“labels” here includes every text that can have an associated accesskey; buttons, menu options, etc.), accesskeys and commandkeys. For the labels category, you can also tell MozillaTranslator to consider keys with no suffix, using “[:empty:]” as suffix.

Once you have set the possible suffixes to be taken into account, you can run **Edit | Check keybinding**. You will get in return a list of labels whose associated accesskey uses a character not being part of the label itself, allowing you to correct them.

As you can modify the suffixes anytime, wrong or missing associations may happen in MozillaTranslator database. To correct it, you'll need to run **Edit | Relink labels & keys**. You'll get the dialog on the right. Leave “Keep existing links” checked if you don't want to replace existing links, or uncheck it to perform a full relinking between labels and accesskeys & commandkeys.



Exporting JAR, XPI and SCM

If you are working with a SCM-based product, you can use **Export | SCM-based Translation** to get the same directory structure than the original en-US one, but with the ab-CD content. However, as Mozilla repositories use a different en-US directory structure than for ab-CD localizations, you will need to set “link points” using the “On Export SCM, export to:” field in the Info Panel when a component (folder) is selected. As an example, look at these replacements for browser component, assuming that you set your product base dir in the “mozilla” dir of the CVS repository (the same is true for Mercurial repositories):

en-US component (in CVS and in MT Chrome View)	Value to be set in the corresponding "On Export SCM, export to:" field
/browser/locales/en-US	/[:locale:]/browser
/extensions/reporter/locales/en-US	/[:locale:]/extensions/reporter
/other-licenses/branding/firefox/locales/en-US	/[:locale:]/other-licenses/branding/firefox

If you are working with a JAR-based product (which is no longer the usual case nowadays), you may find useful the **Export | JAR** and **Export | XPI** options. Export JAR allows you to generate a JAR file with the same structure than the en-US original one, but with the ab-CD contents. Export XPI allows you to build a langpack XPI with the JAR files you select (if the product has several JARs defined, like Platform Neutral, Windows, Linux and Mac XPI, region JAR, etc.) and an install.js with the right instructions to install it in a XPFE host (namely, SeaMonkey 1.1.x).

Migrating a product

Let's suppose you have two very similar products, but with some differences. This was the case in the past with Mozilla Suite and Netscape 6.0/7.0. With the **Advanced | Migrate Product** option, you can have both products in your glossary, translate strings in one of them, and then migrate the translated strings to the other one, just where this latter one has them in the same path, file and key. Probably not as useful now as in the past, but still nice to have it.

Working in group

MT doesn't work in a on-line way, but still provides some basic functionality for teams sharing the translation workload. All members should have the same starting glossary with the full product to translate, and if it is based on a SCM system, all they should try to keep it up to date.

With the previous starting premises, a coordinator may "assign" branches in the glossary to each member (branches can be at any level, just be sure to not assign any member a branch which contains inside a "sub-branch" assigned to other one). Then, after translating the assigned section, and by using **Export | Partial glossary** and **Import | Partial glossary**, sections of the glossary can be handed back to the coordinator to merge them back.

Sometimes the translation contains really huge files (built-in help files are a nice example); in this case, a similar procedure can be followed with individual files (**Export | File** and **Import | File**).

Keeping your work up to date

OK, so you have translated a Mozilla application to your language (probably Mozilla Firefox, and probably in the Aurora channel). Now new strings have arrived to the Aurora channel and it is time to update your translation. What should you do?

It is easy, don't worry. Just follow these steps:

1. First of all, update your en-US repository (change to the base directory where the repository lives, and type `hg -v pull -u`, or `svn update`, or whatever applies to your specific SCM tool).
2. Enter MozillaTranslator.
3. Run **File | Update Product**.
4. Localize the list of strings that will appear after updating.
5. Run automated tests (**Edit | Automated Test Queries | All above at once**) to find out potential errors.
6. Export your products (**Export | SCM-based translation**).
7. Create a language pack to test your changes.
8. Do the corrections you need opening MT and searching (**Edit | Search**) for strings.
9. Commit and push your ab-CD repository.

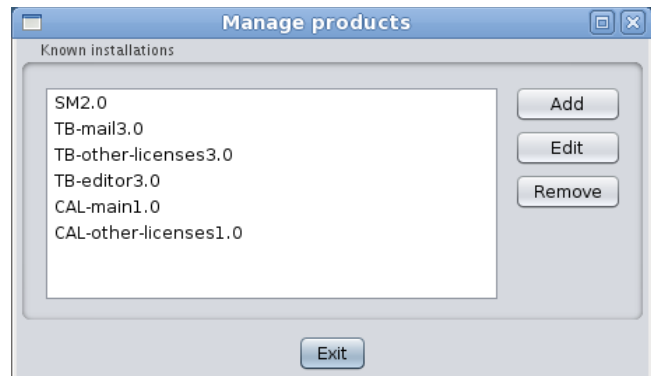
Menu Reference Guide

File Menu

Manage Products

Used to create and remove products, and to modify their properties.

- To add a product to the glossary, click the Add button. You will get a dialog to enter the product properties (name, type, source...).
- To modify the product properties (not the product data to be translated), select a product in the list and click the Edit button. The Edit Product dialog will be shown.
- To remove a product, along with all of its data (both original and translated strings) and properties, select a product in the list and click the Remove button. You will be asked for confirmation.

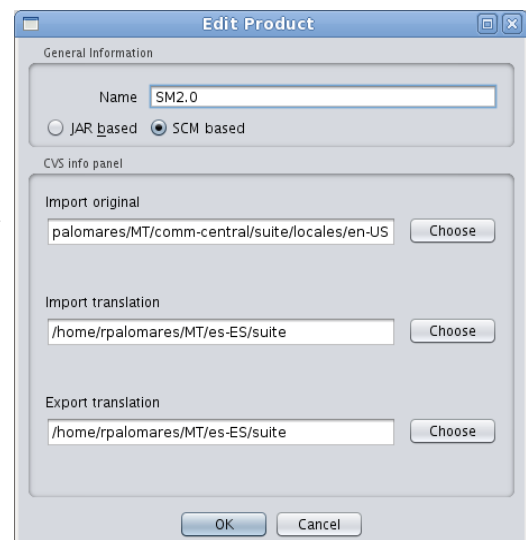


The Edit Product Dialog has these options:

- Name: a short name for the product. It is used just inside MT for your own reference.
- JAR based / SCM based: select JAR based if the original en-US strings are packed in a JAR file, SCM if they are laid out in a regular directory structure (which may or may not have been retrieved through a checkout from a repository; it could even be the expanded content of a JAR file).

SCM info panel

If you select SCM based, the following data will be presented (unlike JAR based, this info is not needed to work with MT, but still you will want to fill it here or else you will need to fill the same data across different MT screens):

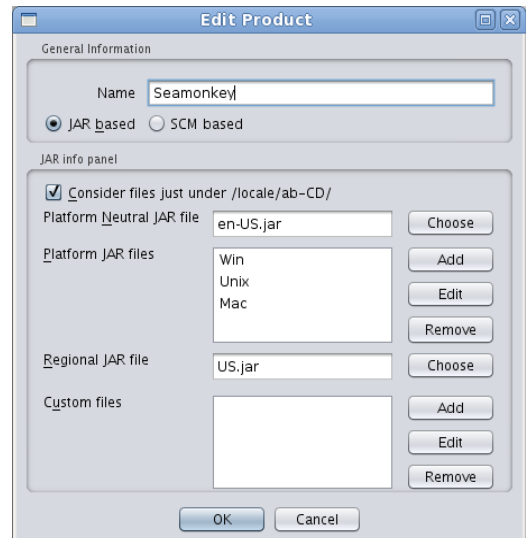


- Import original: the base path where original en-US files live.
- Import translation: the base path where translated ab-CD files live, maybe from previous works.
- Export translation: the base path where to put translated ab-CD files when running **Export | SCM-based Translation**. It is likely that you want this and the previous path to be the same.

JAR info panel

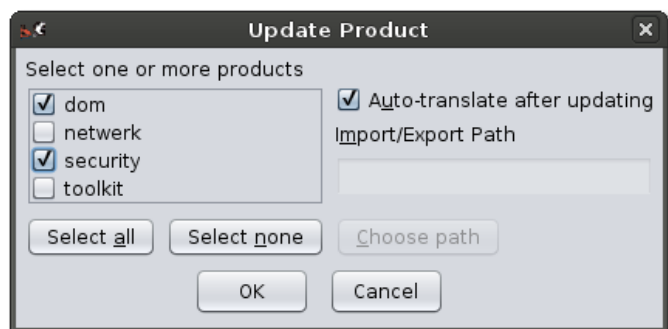
If you have set the product to be JAR based, other options will be shown:

- Platform Neutral JAR file: the file with platform independent (i.e.: the same regardless of it running on Windows, Unix or Mac) locale files. You will always use this locale.
- Platform JAR files: Mozilla Suite, SeaMonkey and maybe other products define platform-dependent JAR files with strings specific to the underlying OS.
- Regional JAR file: some localizable content that may vary depending between regions, even if all regions use the same language. When packed into a XPI this is called the content/region pack.
- Custom files: unfortunately, this option doesn't do anything useful. When it get fixed, you will be able to add files not present in the original JAR files and export them. Each custom file will ask you for three data:
 - Name: name that will be shown in the Chrome View tree and in the File column in the table views.
 - Original file: path in your file system to the file to be inserted.
 - Mount point: path inside of the exported XPI or JAR.



Update Product

This option presents a dialog in which you must select one or several products to be rescan for changes (new files and strings, modified strings, and deleted files and strings). For JAR based products, MT will look for JAR files



defined in the product properties; if the product is SCM-based, the SCM import path of the product will be used.

If you select just one product and it is SCM-based, you can change the import path in the import/export path field, which will be automatically filled with the data from the product properties.

Once you confirm the dialog, you will be asked to choose the columns to be presented in the results table. For more info, see “Choosing columns” in the Edit menu. After that, the products will be rescanned and you'll get a list of new and modified strings.

Save Glossary

All data managed by MT during its execution is loaded in memory. You may want to save the glossary once in a while during long sessions, to minimize risk of data loss in case of power outage, program or OS hangs, etc.

Starting from MT v5.25, prior to saving the glossary, the previous Glossary.zip file is renamed to Glossary.bkf. Since MT v5.30, you can split each product information in a separate file inside Glossary.zip, to reduce memory requirements while saving it.

As this process is the one with the biggest memory requirement in MozillaTranslator, a special exception handling has been added so if the program runs out of memory while re-creating the ZIP file, it will still save the data in a glossary.txt file.

Save Settings

In a similar way to the previous one, you can save MT preferences selecting this option to avoid risk of data loss.

Setup

This option allows you to set several MT preferences. Many of them should be reviewed and adjusted to get the most of MT. The Setup dialog has these tabs and options:

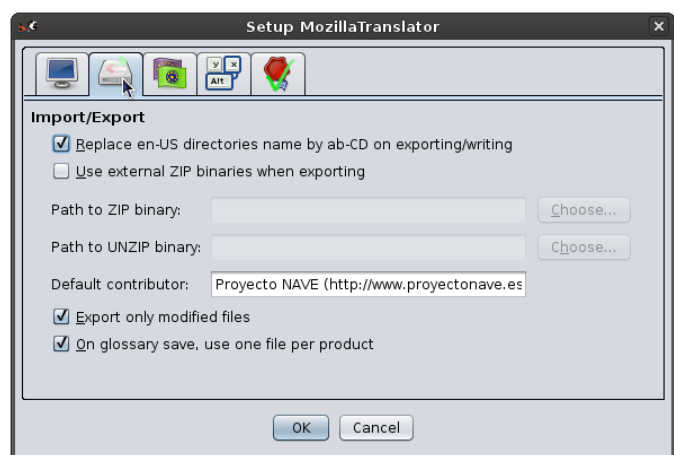
Display Tab

- Font for Edit Phrase dialog, Font for table views: some non-Western languages may have problems using the default DialogInput and Dialog fonts, so here they can specify what font family, size and style to use.
- Look And Feel: you can choose between several Java Look And Feels, which are kind of skins.
- Make ShowWhat dialog visible: check to get the “Select which columns” dialog every time a table is to be presented. Uncheck if you are happy with your last column set selection.

Input/Output Tab

This tab gathers some options useful for import and export functions.

- Replace en-US directories name with ab-CD on exporting/writing: this is done automatically for JAR files, but you may want to do the same for **Export | SCM-based translation**.
- Use external ZIP binaries when exporting: on Unix/Linux, Java built-in ZIP/JAR management classes fail to provide the right permissions in the files added to the archive. This can be solved by just uncompressing and re-compressing again the JAR/XPI files invoking the standard InfoZIP binaries, which are available in most Linux distributions. This is not needed in Windows at all.
- Path to ZIP binary: if you switch on the previous field, you should supply the path to the ZIP binary.
- Path to UNZIP binary: path to the UNZIP binary in case you selected the Use external ZIP binaries when exporting.
- Default contributor: here you can write the localizer credit to be inserted in license files under the “Contributors” section of the license. Keep in mind that MT is no so smart to preserve previous license in the translated file; it will just grab the en-US license and append the value in the translated file at export time.
- Export only modified files: if you set this option, when you choose **Export | SCM-based translation** MT will only dump modified files since last export. Although MT output is consistent, meaning that two consecutive exports of the same file will produce the same output, you may want to check this option. You can still export all the files whenever you wish, as the export dialog replicates this option.



- On glossary save, use one file per product: Glossary.zip used to contain a glossary.txt file with all the data related to products, files, original strings, translated strings, etc. Creating this file in memory and putting it into Glossary.zip could cause OutOfMemoryException errors, so a change has been made to split the information of each product in individual files, all of them added to Glossary.zip. This reduces the memory consumption, so it is a good idea to check this option.

Translation Assistance

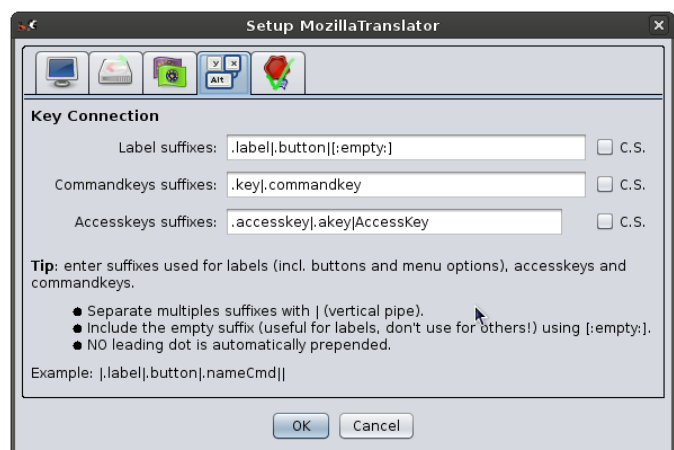
This tab contains the preferences aimed to ease the translation process. Currently, this covers translation suggestions, with three preferences:

- Use translation suggestions on Edit Phrase Dialog: check this option to enable translation suggestions while in the Edit Phrase dialog. If enabled, you will get a visual indicator when MT can offer you suggestions. See the Edit Phrase dialog section on the Edit menu for details.
- Provide suggestions for XX% coincidence or more: MT will calculate the match percentage (computed using Levenshtein distance algorithm) of potential translation suggestions and include only those equal or above the value set in this preference.
- Auto-translate on Product Update: you can make MT to pre-fill empty strings with the best possible translation suggestion (respecting the previous percentage). This will save you a lot of time, as you may just need to confirm or change MT proposed translations.

Key Connections Tab

Here you can define suffixes for labels, accesskeys and commandkeys. Separate suffixes using “|” (pipe). As most moderns graphical programs, Mozilla allows access to most of its options by both mouse and keyboard. Accessing through the keyboard is done in two ways:

- Keyboard shortcuts available through the whole program, or most of it. Usually they involve pressing [Ctrl] (or the equivalent Macintosh key) plus another key, and they are unique (the same combination always behaves in the same way inside the program). This kind of shortcuts are called “commandkeys”.



- Keyboard quick access keys available in menus and dialogs. They are represented as an underlined character of the UI label associated to the function. Sometimes they need to be invoked using [Alt] plus the underlined character. The same character is used in different dialogs, and sometimes even it appears several times in the same UI (although it is something to be avoided). These are the so called “accesskeys”.

The comandkeys need not to have their character in common with the function they perform. For instance, you use [Ctrl]+[X] to cut text to the clipboard, and yet the menu option is **Edit | Cut** and thus doesn't have an X in their label.

However, that's not the case with accesskeys. The **File | Open...** accesskey is O, but when you translate it into (for instance) Spanish, you have “Abrir...”, so if you keep O as the accesskey, the menu option will be shown in the Mozilla UI as “Abrir... (O)”.

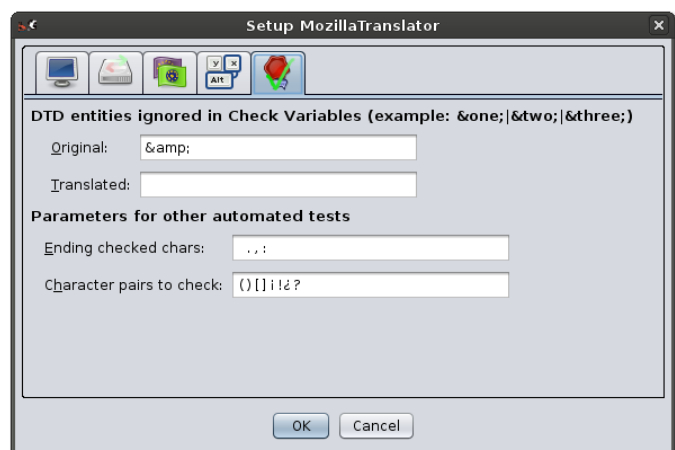
All of this means that a connection must exist between UI labels and (at least) accesskeys, and that you need to adapt accesskeys to the label translation.

There are rules managing how to name strings that act as labels, accesskeys and commandkeys, mainly by adding suffixes for every of them. And, as every rule, they are sometimes not followed. :-) So MT v5.1 introduced some user preferences to let the localizer set a number of possible suffixes for labels (“labels” here includes every text that can have an associated accesskey; buttons, menu options, etc.), accesskeys and commandkeys. For the labels category, you can also tell MT to consider keys with no suffix, using “[:empty:]” as suffix.

Automated tests Tab

In this tab you can supply values for these options:

- Original DTD entities ignored in Check Variables: “Check Variables” is a query that looks for strings that have different entities in original DTD content than in translated ones, as this could mean an error in translation. However, sometimes you will



want MT to ignore certain entities, like & in en-US. In that case, enter the entities to be ignored in this field separated with “|”.

- Translated DTD entities ignored in Check Variables: complementing the previous one, you can enter here DTD entities to be ignored for translated content.
- Ending checked chars: you can enter here a list of characters that, when any of them appears ending an original string, will be checked to be also the last one in the corresponding translated string (See Check for different endings at Automated Test Queries).
- Character pairs to check: also in MT v5.24, a new test has been introduced that looks for mismatched characters (characters that usually appear in pairs, like “(“ and “)”). Here you can provide the pairs you want MT to look for. To know more about the feature (which needs polishing, to be honest), review the Automated tests section.

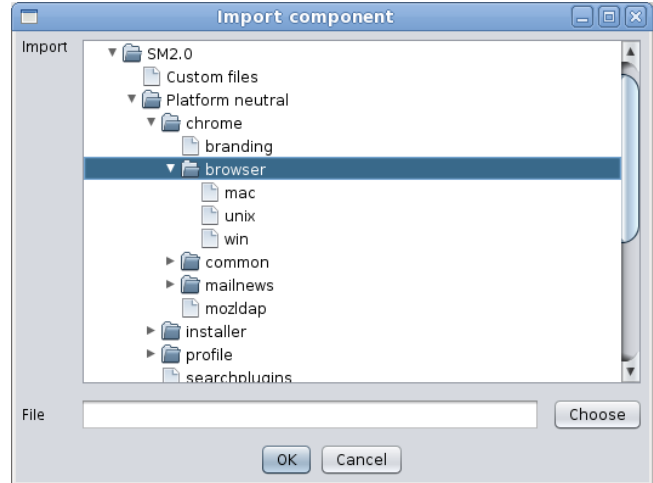
Quit

This option exits MT. Before shutting down the program, you will be asked to save the glossary and settings.

Import menu

Import Partial Glossary

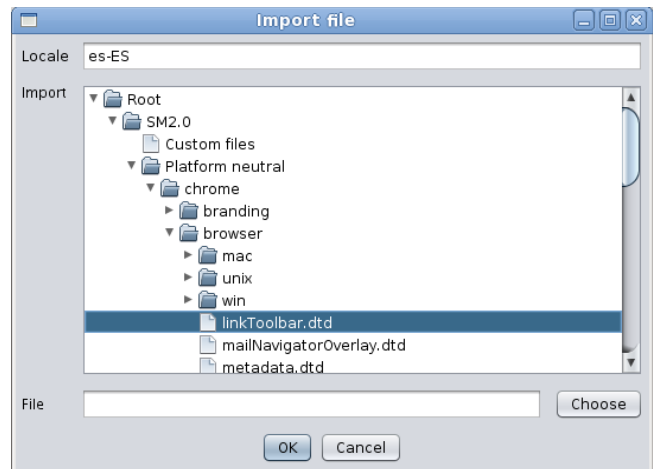
This option allows you to import a previously saved partial glossary and merge it into the main glossary. You will have to choose a component in the import tree and the file containing the partial glossary. And, differently to MT 5.0x and previous versions, you don't need to specify a leaf node. In the image, the "global" node, which has "history" and "xpinstall" under it, can be selected. The partial glossary is expected to have the same structure than the one you select in the dialog. This option is useful for collaborative work.



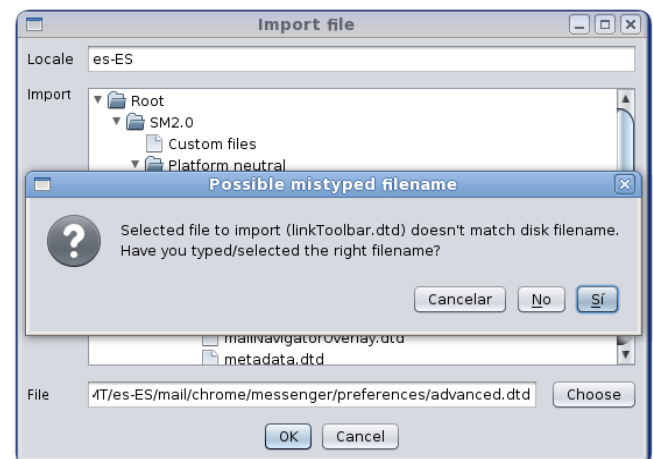
Import File

Sometimes, you want to just import a single file instead of a bunch of them. Import file allows you to do this. The dialog structure is the same than for importing partial glossaries, except that in this case you must choose a leaf node in the tree, one belonging to a file, and that you need to specify the locale to export.

In the image, the file "linkToolbar.dtd" has been selected. Note that, in the previous image from Import partial glossary, the "mac", "unix" and "win" nodes were leaf nodes, while in this image they are folders (since they are components, not files).

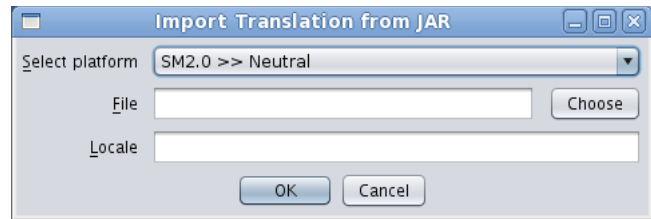


If you happen to type or choose a file in your disk whose name doesn't match the name of the file node selected in the "Import" tree, you'll be asked to confirm the operation.



Import Translation from JAR

If you are working with a JAR-based product, you may import the content of a previous translation in the form of JAR files, like the one shipped as actual translations. So, if you have no access to the Glossary.zip file for a translation but have access to the resulting JAR files, you can use these to avoid having to restart from scratch. In this case, select the product and platform from the drop down list, the JAR file (ab-CD.jar, ab-win.jar, CD.jar, etc.), and the right locale code in which import the translation.



Note: keep in mind that the meta data (Keep Original, Fuzzy, comments, etc.) cannot be retrieved from the JAR files.

Import Translation from SCM

On the other side, if you are working with a SCM-based product, you may find yourself in a similar situation, except that you will have an outdated SCM repository (outdated because nobody has brought the content in pair with the en-US, not because the SCM software is obsolete itself). The SCM import dialog lets you select one or several SCM-based products to import. If you select just one product, you can also change the SCM import path.



Note: keep in mind that, like in the JAR case, the meta data (Keep Original, Fuzzy, comments, etc.) cannot be retrieved from the raw files in SCM.

Export menu

Export Partial Glossary

This option is the reverse of Import partial glossary. It lets you to select a component and export it to a file that can be later imported. Also like in Import partial glossary, and also differently to MT 5.0x and previous versions, you don't need to specify a leaf node.

Export File

This option is the reverse of Import files. You must choose the locale to export and a file in the tree, and write or select a path and filename where to export. As in the import, you'll be warned if the selected node name in the tree doesn't match the actual filename in disk where to export it, to avoid mistakenly overwrite other files.

Create XPI File

This option creates an installable language pack for XPFE-based products (i.e., SeaMonkey \leq 1.1). It is not useful for Firefox, Thunderbird and, in general, XUL/Toolkit based products, so expect it to disappear in future versions. These are the fields:

- Locale: the locale to be exported.
- File: the path and file name where to save the XPI.
- Display Name: the string that will appear in the langpack/extension description.
- Author: the author name. You can use a team name if you want, since it is just a descriptive field.
- Preview URL: this actually doesn't do anything useful. It is a descriptive field in install.js files.
- Version: this field is disabled for edition. It will have its value taken from the selected product in the table "Select platforms and regions" below.

Product	Version	Neutral	Windows	Unix	Mac	Region	Custom	Check All
Seamonkey	1.8.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

- Use external ZIP/UNZIP: check this if you have configured ZIP/UNZIP paths in MT preferences, to workaround permissions problem in Unix/Linux systems.
- Select platforms and regions table: for every product created in the system, a row will be presented in the table in the dialog. You can check any cell you want to be included in the XPI, but keep in mind that MT won't let you select check boxes from two products having different versions. Checking/unchecking the "Check All" column checks/unchecks all the cells in the same row, as a quick way to choose a whole product.

Localizing the XPI installation messages

The resulting XPI will use a `install.js` provided by Robert Kaiser which allows the user to install the langpack global or locally (in the user profile). The messages presented to the user during installation are located in a file named `install.properties` put in the root of the XPI archive (which is actually a ZIP-based archive). If you want to provide translated messages, follow these steps:

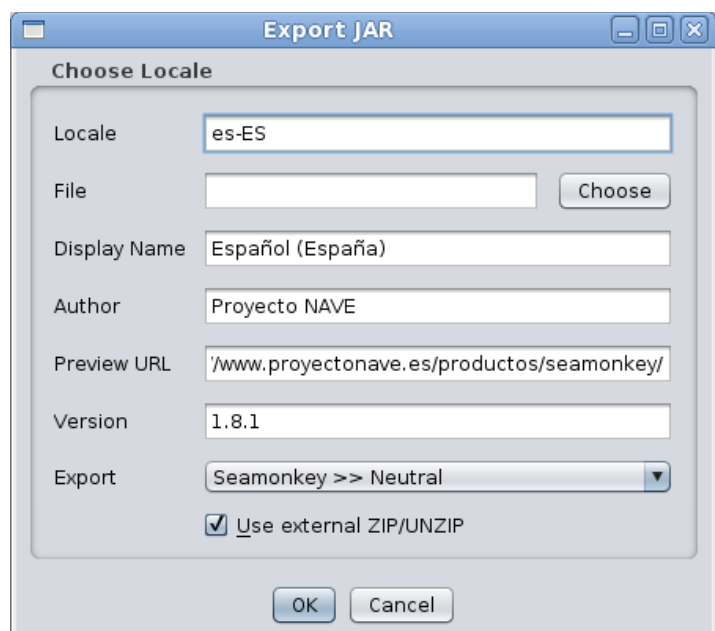
1. Open the `mt5xx.jar` with a ZIP extracting/browsing utility.
2. Locate the file `org/mozillatranslator/resources/install.properties`.
3. Extract that file to the same directory where `Glossary.zip` is.
4. Translate it with an UTF-8 editor (you can use MT for this if you want, but it may be too much hassle).

If you don't provide the translated file, the en-US version will be used.

Create JAR File

This option creates JAR files with translated locale files. You will have to provide these values:

- Locale: write the locale ab-CD code for your language.
- File: write or select the filename where to save the JAR file.
- Display Name: the description of the JAR file content.
- Author: the author of the translation. It is just a descriptive field.



- Preview URL: not used in practice. Again, it is just a descriptive field.
- Version: this field is disabled; its value will be taken from the platform taken in the Export field.
- Export: a drop down list with every product and platform defined in the Glossary.zip.
- Use external ZIP/UNZIP: check this if you have configured ZIP/UNZIP paths in MT preferences, to workaroud permissions problem in Unix/Linux systems.

Export in XML Format

This sub-menu contains options to export parts of the translation as XML files. These files are not intended to exchange data with other MT users, but to obtain that data in a format easily manageable.

...File

In this option you will choose a file in the tree, specify a locale code and provide a filename. The dialog is very much the same than the regular export file one.

Note: keep in mind that content of binary files will not be exported. If you choose a binary file, the XML you will get will only contain the meta data and general information of the file.

...Component

In this option you will choose a component in the tree and provide a filename. Again, the dialog is similar to the regular export glossary one.

...Glossary

In this option you will have to provide just the filename where to save glossary in XML format.

Export SCM-based Translation

This option lets you to export one or several SCM-based products reflecting the directory structure. If you have set the "Replace en-US directories name by ab-CD on exporting/writing", the directories named "en-US" will be renamed to your locale during export. This alone won't be enough to convert the export to the right structure for committing it to the L10n repository, as the directory layout is completely different, but you may workaroud this (for more information, see "The Info Tab" section of Chrome View, in the Edit Menu).

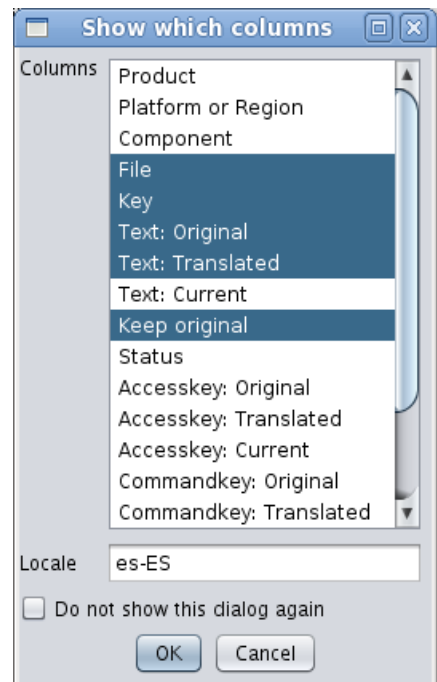
Edit Menu

Chrome View

Chrome View is the most complex and complete view in MT. It lets you browse every product, component and file in the Glossary.zip. It is more useful when you are doing the first translation of a product (ie., you are not reusing a previous translation), and for getting a view of a full file, instead of seeing only new or modified strings of it.

The Show What dialog

As with every view of the data, first you will be presented with the Show What dialog. In this dialog, you choose which columns you want to be in the table view. Select the columns to be shown by [Control]+clicking each name, and insert the locale code under the selection list, which needs to have at least two characters. If you are pretty sure you will always want to have the same column set shown, you can check “Do not show this dialog again” (it can be restored through MT setup dialog, in case you need it).



Consider each file, and each string inside, in the internal database is placed under a well defined hierarchy:

/ Product / [Platform | Region | Custom] / Component(0..n) / File / Key

The previous hierarchy means that, under a Product, you'll find either a platform (including “Platform neutral”), a Region or a Custom container. Under each of these, you'll find a number of successive components, starting at 0 (no component; that's the case for custom containers, and it may happen too in some SCM-based products, depending of what you define as the base directory). Under that, you'll find several files with keys inside them. Each key element belongs to a single string and has several properties:

Column name	Description
Key	The name referenced by the application for this string. If it shows in boldface , then the key has a localization note associated to it (you will be able to see it through the Edit Phrase dialog).

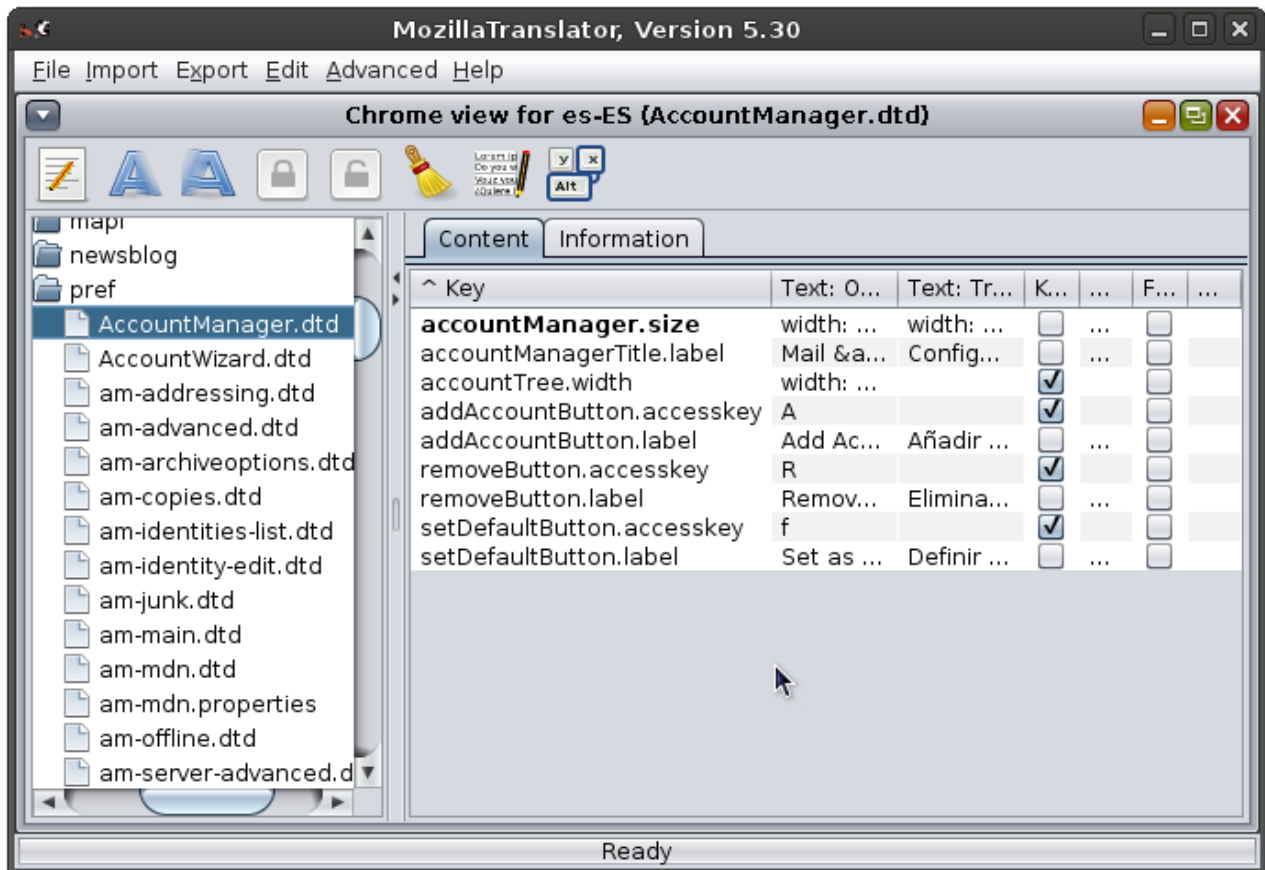
Text: Original	The original string itself to be not localized (probably in English).
Accesskey: Original	The keyboard shortcut for a menu item or an UI widget currently shown in the open window (Alt/Apple+letter). This is the letter of the Text: Original string will be underlined. (a.k.a. accelerator key).
Commandkey: Original	The application wide keyboard shortcut for the function indicated by Text: Original (Ctrl+letter).
Text: Translated	Same as for the three 'Original' counterpart, this time the fields can be edited, entering the localized version for the currently selected locale.
Accesskey: Translated	
Commandkey: Translated	
Text: Current	These fields don't really exist in the database. They just show what the program would export for the current locale; e.g., when the Text: Translated field is empty the program will export the Original content.
Accesskey: Current	
Commandkey: Current	
Keep Original	This check mark box, when flagged, disables the 'Translated' fields, and take the current values from the 'Original' fields.
Status	This is mostly a useless field.
Comment	A text field to enter relevant comments. The comments are only saved if an actual translation exists (ie. marking a key as Keep Original does not allow to save a comment for that key).
Fuzzy	This field marks <i>non-reviewed</i> strings.
Result Comment	After running most automated test queries, some strings can have a "result comment" attached to them to better explain why they have been selected in the query.

The Chrome View Window

The above image shows a typical Chrome View. It has the following areas:

- The tool bar.
- The tree.
- The content tab in non-image files.
- The content tab in image files.
- The information tab.



Each area is explained in the following pages.












The tool bar



The tool bar allows you to perform some operations with the string or strings selected in the view. The function of each button is the following:

- 
 Edit selected phrase: launches the Edit Phrase dialog, which presents a more complete view of the string and its status. From that dialog you can go back and forward through the file strings.
- 
 Edit image file: this button is not present in Chrome View, but it is in other table views. It shows the Edit image file when the single string inside image files is selected and works exactly the same than Content tab when an image file is selected in the tree.

- 
 Copy to all: this button is not present in Chrome View, either, but it is in Redundant Strings query. It allows to set every row with the same original with the same translation than the one in the string selected.
- 
 View in Chrome View: this button is not present in Chrome View, but it is in other table views. Given a selected phrase in any table view, it opens a Chrome View with that phrase focused. This way, you can see the phrase *in its context*, inside the file where it lives, with the full component path to it.
- 
 Fuzzy: activates the Fuzzy flag in the selected phrase(s). It is specially useful if you want to mark as Fuzzy a bunch of strings; just select the desired strings (usual [Shift] and [Ctrl] key combinations apply for selecting a consecutive / non-consecutive group) and press the button.
- 
 Unfuzzy: deactivates the Fuzzy flag in the selected phrase(s).
- 
 Keep Or.: sets the Keep Original flag in the selected phrase(s).
- 
 Unkeep Or.: unsets the Keep Original flag in the selected phrase(s).
- 
 Clear trns.: clear the translation in the selected phrase(s). Be careful, there is no Undo action for this!!
- 
 Mass change: this button opens a dialog which allows you to perform one or several changes to the selected phrase(s). It includes the features of the previous buttons (clear translations, fuzzy, unfuzzy, etc.) and others, like changing the Translation Status field. There is no Undo action for this, so you will be asked to confirm the changes before applying them.
- 
 Auto-assign accesskeys: perform an automatic accesskeys character assignment based on predefined rules. A new feature of MozillaTranslator v5.25 explained below.

The tree

The tree in the left lets you navigate through every product, component and file present in the Glossary.zip. When you click a file, the right side will show its contents; when you click a component or product, no content will be shown, but it still can be useful to obtain some info of it.



The Content tab in non-image files

Last, but not least, the right side shows you two tabs: Content and Information. The Content tab shows the table view (except if the file is an image, see below) with every selected column. In the image, note that the first entity key name is in bold, because it has a localization note attached to it. To see the localization note, you have to select the string and go to the Edit Phrase dialog.

^ Key	Text: Original	Text: Tran...	Keep	Fuzzy
accountManager.size	width: 55em; ...	width: 65...	<input type="checkbox"/>	<input type="checkbox"/>
accountManagerTitle...	Mail & N...	Configurac...	<input type="checkbox"/>	<input type="checkbox"/>
accountTree.width	width: 17em;		<input checked="" type="checkbox"/>	<input type="checkbox"/>
addAccountButton.ac...	A		<input checked="" type="checkbox"/>	<input type="checkbox"/>
addAccountButton.la...	Add Account...	Añadir cu...	<input type="checkbox"/>	<input type="checkbox"/>
duplicate.label	Duplicate	Duplicar	<input type="checkbox"/>	<input type="checkbox"/>
duplicateButton.label	Duplicate	Duplicar	<input type="checkbox"/>	<input type="checkbox"/>
macAccountManager...	width: 60em; ...		<input checked="" type="checkbox"/>	<input type="checkbox"/>
removeButton.access...	R		<input checked="" type="checkbox"/>	<input type="checkbox"/>
removeButton.label	Remove Acco...	Eliminar c...	<input type="checkbox"/>	<input type="checkbox"/>
setDefaultButton.acc...	f		<input checked="" type="checkbox"/>	<input type="checkbox"/>
setDefaultButton.label	Set as Default	Definir co...	<input type="checkbox"/>	<input type="checkbox"/>

Note: if you need to type an ellipsis (...) in the Translated column (i.e, while you are translating), just press [Alt] + [.] and you will get it!

Starting from MT v5.30, you can find a colorful results view. The reason is that Translation Status values have associated colors that help you to identify strings with automated translations or, in general, requiring review.

^ Key	Text: Original	Text: Translated	Keep	Status	...
accnameDesc.label	Enter the name by which y...	Introduzca el nombr...	<input type="checkbox"/>	Translated	<input type="checkbox"/>
accnameLabel.accesskey	A		<input checked="" type="checkbox"/>		<input type="checkbox"/>
accnameLabel.label	Account Name:	Nombre de la cuenta:	<input type="checkbox"/>	Untranslated	<input type="checkbox"/>
accnameTitle.label	Account Name	Nombre de la cuenta	<input type="checkbox"/>	Modified	<input type="checkbox"/>
accountSetupInfo2.label	In order to receive messa...	Para poder recibir m...	<input type="checkbox"/>	Approximated	<input type="checkbox"/>
accountTypeDesc2.label	This Wizard will collect the ...	Este asistente recop...	<input type="checkbox"/>	Proposed	<input type="checkbox"/>
accountTypeDirections.label	Select the type of account...	Seleccione el tipo de...	<input type="checkbox"/>	Copied	<input type="checkbox"/>
accountTypeMail.accesskey	m	c	<input type="checkbox"/>	Translated	<input type="checkbox"/>
accountTypeMail.label	Email account	Cuenta de correo el...	<input type="checkbox"/>	Translated	<input type="checkbox"/>
accountTypeNews.accesskey	w	n	<input type="checkbox"/>	Translated	<input type="checkbox"/>

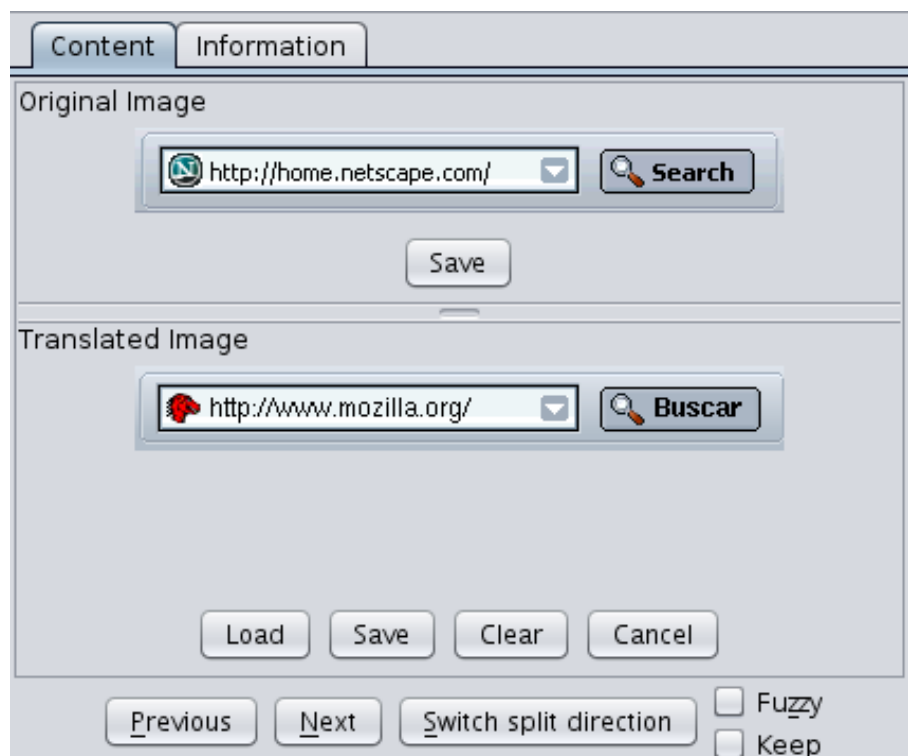
The possible values, their associated color and its meaning are the following:

Note: in future versions, MT will allow to customize the colors bound to every Translation Status.

- (Empty): empty Translation Status value means the original string is set as Keep Original, which most of the time equals to being translated, so there is no special color associated to this value.
- Untranslated (red): this string is not set as Keep Original and has no provided translation.
- Modified (orange): the original string has been modified since it was last translated. Currently this value is not used by MT auto-translate feature; you can set it manually, though, if you need it.
- Approximated (light orange): the translation value has been filled with one non-100% match suggestion.
- Proposed (light green): the translation value has been filled with one from several 100% match suggestions list.
- Copied (intense green): the translation value has been filled with the only 100% match existing suggestion.
- Translated (white): the translation has been set as translated.

The Content tab in image files

If you are viewing an image file, the Content tab changes to this view. The upper section shows the original (en-US) image, which you can save if you want to use it as starting point for a localized version of it. The bottom section shows the localized image, if any. You can load, save or clear the translated image. You can also cancel the last action done on the translated image. Besides that, you can set/unset the Fuzzy



and Keep flags, and move back and forward to the next file. Lastly, as some images are tall and others wide, you can switch the split direction.

The Information tab

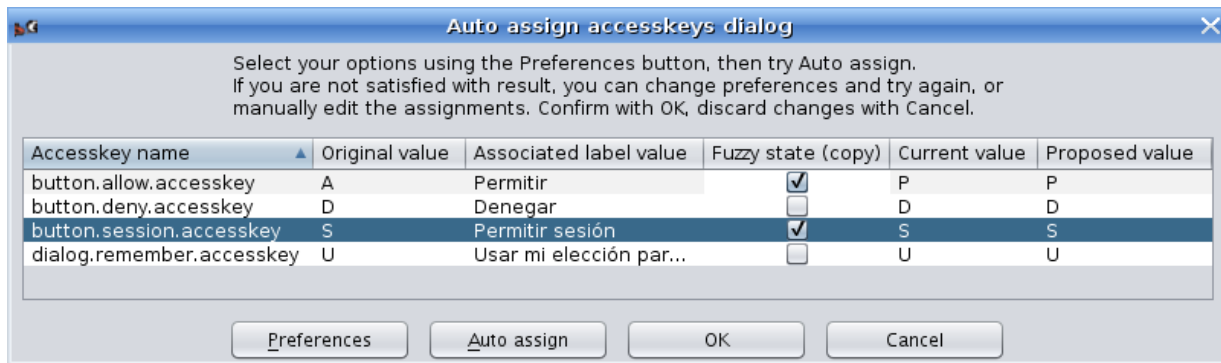
The Information tab shows general information about the selected node in the tree at the left. It also lets you do two things:


- set an alternate export directory for the selected component, so when you choose **Export | SCM-based Translation**, you can get the right layout for L10n repository. In the “Exporting JAR, XPI and SCM” of the previous MT User Guide, you will find some samples.
- mark a file **not** to be exported. This is useful for some productization files, which typically happens in Mozilla products with searchplugin files and region.properties. Why would you avoid exporting a file? Well, the main reason is because you absolutely must not to make changes to these files without getting prior approval from Mozilla L10n (by filing a bug). Besides that, you will also want not to export them because the number and name of strings in it won't match the original one (this may happen with region.properties) or because the file is not needed/wanted at all (like google.xml searchplugin or any other searchplugin not included in your localization).
- get some statistics information by pressing the “Calculate” button. The interesting thing about the statistics is that you can get them on any node (except the Root node).



The new Auto-assign Accesskey Dialog

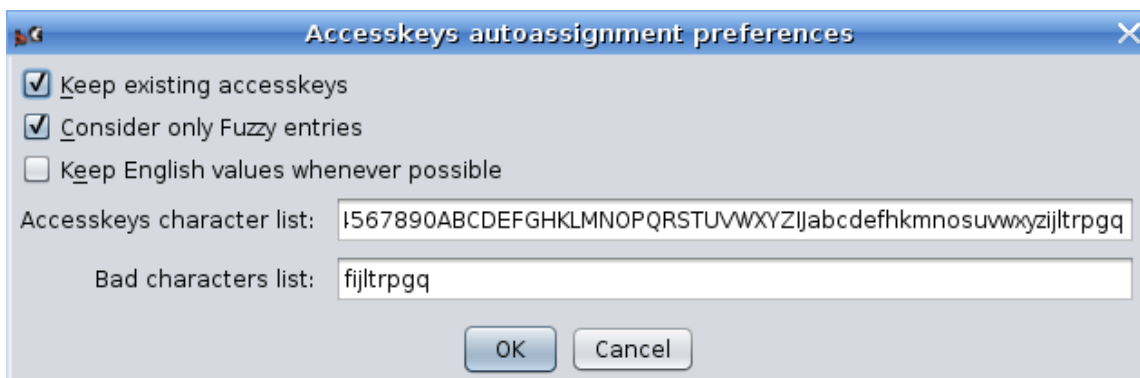
If you have been localizing Mozilla products for a while now, you, like me, probably hate having to choose between available characters for accesskeys. MozillaTranslator can't fully automate this task for multiple reasons, but it can somewhat help you. You just have to follow these steps:



1. In the Chrome View for a specific file, select the rows with the accesskeys you want MT to (re)assign automatically. You may include rows which are not accesskeys as MT will detect them and filter it out. Hint: if a L10n file contains only elements from a single dialog, you can focus any row in the Chrome View and hit [Ctrl]+[A] to select all rows (MT will discard non-accesskeys items).
2. Click the Auto-assign accesskey icon to show the Auto-assign Accesskey Dialog. 
3. Adjust preferences (if needed) by clicking the Preferences button.
4. Adjust Fuzzy state (if needed) for the desired rows.
5. Click Auto-assign button to see MT's attempt.
6. Correct MT assignments if needed.
7. Click OK to copy the dialog new values to the Chrome View.

Auto-assign Acceskeys Preferences Dialog

To access the Auto-assign Accesskeys Preferences Dialog, click the Preferences button in the Auto-assign Accesskeys Dialog. You will see this dialog:



- Keep existing accesskeys: if you check this, any non-empty accesskey will be preserved. Usually you may want this setting on.

- Consider only Fuzzy entries: select this to only consider accesskeys with the Fuzzy flag set (which is automatically set for new/modified entites/keys when updating a product in MT).
- Keep English values whenever possible: if you try to keep the same accesskey for English in your language whenever the translated label includes the chosen accesskey in English, set this option on. For instance, if a label with the "Remove" value has "R" as accesskey, and the localized label value (to Spanish) is "Eliminar", setting this option on will mean that "r" will be chosen as the localized accesskey.
- Accesskey character list: enter here the full list of characters that may be used as accesskeys in your language, sorted by preference to be chosen. Note that capital and regular letters must be listed separately; this way you can tell MT that "R" is good, while "r" is not so good for accesskey values (as it is too narrow).
- Bad characters list: enter here the list of characters that should be used only as a last option. A character may end up in this list because it being narrow, or because it has a lower line that may get mixed with the accesskey underscore indicator.

Once you are happy with the values, press OK to accept the dialog.

How it works

Having an overview of how auto-assign accesskeys operates may help you to use it. So here it is what it does and in what order:

1. The list of available characters is purged from already assigned values (in case "Keep existing accesskeys" is set).
2. If "Keep English values if possible", then a loop is run to assign localized empty accesskeys to their English original values when possible (ie., when doing so does not lead to accesskeys shown between parenthesis, and the English value is not in the "bad character" list). The list of available characters is correspondingly adjusted.
3. A loop is run for each available character (and NOT for each pending accesskey). For each character, a match is tried among pending accesskeys and, if found, it is assigned to the accesskey, and the available character list is adjusted, removing both the uppercase and lowercase letter, or the digit.

Some use cases

To get the most out of this feature, the key is to select a meaningful set of accesskeys (both assigned and pending) as close as possible to the actual set of accesskeys in a dialog or menu. There are different scenarios:

1. In some cases, a whole file represents a single dialog, so it is best to select all entities in it (select a row, then press [Ctrl]+[A]). This way, MT will know which characters are already in use for the dialog and choose

others among the still available. Typical examples of this scenario are preferences dialogs. To access to the whole file of a new/updated accesskey from the Update Product result list, select the accesskey and click the “Show In Chrome” button.



2. In other cases, a single (huge) file contains a lot of UI elements, including many menus and submenus. Hopefully, the entity names will help you to identify submenus or other UI containers that you need to select to give MT a context in order to create unique accesskey combinations. Files like browser.dtd, messenger.dtd or calendar.dtd are typical examples of this case.
3. Worst cases are overlays. Overlays are menu elements that can be used in different contexts and menus. For instance, Cut, Copy and Paste are common options used in Edit menu, but also in context (right-click) menus for a selected text, or for an image in a web page or an email. Since these common options combine with other specific menu options and are defined in independent files, you won't be able to select all relevant UI labels and accesskeys at the same time, and therefore MT won't have enough context to propose an accesskey combination with no collisions. The editMenuOverlay.dtd file in Toolkit is an overlay file.

Automated Test Queries

This menu contains a number of queries aimed to provide some basic level of automated QA (Quality Assurance). They all work in the same way: you choose the option, select which columns should be shown, and you get a table view window with the matching strings.

Untranslated Strings

All strings should be translated. Those not requiring translation should have “Keep Original” flag set. This way, running this query will report you with the remaining strings to translate.

Check Keep Original with trns.

This query reports all strings with Keep original flag set having a translation. As the Keep original flag prevails over any translation, all the listed strings should have an empty translation, or otherwise you may want to revise it (either no translation is needed, or Keep original flag should be removed).

View Original == Translated

Unless you are maintaining two or more different translations at once for the same product, you should use the Keep original flag to mark strings not requiring translation, and thus no strings should have a translation that equals to the original string.

Why can't you use Keep original if you maintain two or more translations? Because Keep original is bound to the phrase, not to the translation, so you can't say, for instance, that “No” translates to “Nein” in German, and at the same time “keep it original” for the Spanish translation (which is “No”).

Check Variables

Some strings in properties files include *variables*, in the form %S or %n\$S. At runtime, the Mozilla product substitutes these variables by actual values (for instance: “%S files saved” could print as “3 files saved” at runtime). %n\$S, where n is a number between 1-9, allows the translator to alter the substitution order (for instance: “%1\$S messages were found across %2\$S folders” could be translated as “A lo largo de %2\$S carpetas (folders) se han encontrado %1\$S mensajes (messages)”, and yet the number of messages and folders would be printed in the right position).

DTD files are checked, too. If a original string contains some &entities; the translated string will be checked to have the same entities, too. As sometimes you may need to use entities in the translation not present in the original string and vice versa, you can provide MT with a list of exceptions in the General tab of MT setup.

The Check Variables query searches for original strings with variables and compares if their associated translations have the same variables, returning a list of strings with problems.

Check Key binding

This query gives you a list of strings with problems in their associated accesskeys. More information about this query is available at Labels, commandkeys and accesskeys section in the User Guide.

View DONT_TRANSLATE not honored

Localization notes in DTD and Properties files are written by developers to help localizers to do their job. There is a well defined syntax in localization notes for developers to warn localizers about not translating some strings, or part of them. The localization note must have the string “DONT_TRANSLATE” in it to flag this circumstance. This query looks for strings with localization notes containing “DONT_TRANSLATE” and yet having a translation. Keep in mind that this query has uncovered several strings flagged as DONT_TRANSLATE when they really deserve a translation (at least, in Spanish), so take the results of this query with a grain of salt.

Check for different endings

Usually, at least in western languages, when the original en-US string ends with ".", ":" or a whitespace, so must do it the translated string. However, and especially in the whitespace case, the translator may overlook it and finish the translation with a different character. You can run a check here to catch up these errors. The list of ending characters to be checked defaults to " .,:" (whitespace, dot, comma and colon) and can be customized in the General tab of Setup window, at File menu.

Check for duplicated words

This query looks for two or more identical *tokens* (words or variables), as it usually means a mistake in the translation. There is room for improvement in this query, but still it is already somewhat useful.

All above at once

This query runs all the previous tests at once, giving you the accumulated result. The power of this test comes from the "Result comment" column or the QA Results tab in Edit Phrase Dialog, because every previous test adds useful info in that column/tab, so you can discover and fix in a single query most typical QA problems. The info is cleared for every query run in Edit (even View Binary and View Fuzzy) except Chrome View, and is not saved in the Glossary.zip file.

Check for mismatched pairs

This is a query that looks in the translated texts for characters that usually come in pairs, like "(" and ")", but that are not (one of both is missing). This query creates too many false positives at the moment, though (for instance, it will complain if the translation includes a smiley), so it is not included in the Check All Above at Once accumulated query.

Redundant Strings

This query gives you a list of all original strings that appear more than once in the glossary. It allows you to manually search for different translations for two equal original strings. However, if you need to do this, you'll find more useful the Search for translation inconsistency (explained below).

Search Translation Inconsistency

This query creates a list with strings that match all these criteria:

- The original string appears more than once.

- At least one occurrence of them has been translated in a different way than the others.

Different translations may happen because of these reasons:

- The different contexts surrounding the original strings do it necessary.
- Some of the strings appear in dialogs with size constraints, so some translations need to be abbreviated or otherwise shortened.
- The different translations happened in different moments or by different people. All translations are correct, but it may be weird having the same original string translated in different ways.

For this latter reason, you may want to use this query. Once you get the list, you can sort it by Text: Original to see how the same en-US string has been translated in different ways. If you have such a case and change one translation, all strings with that same translation get affected. For instance:

#	<i>en-US</i>	<i>ab-CD</i>
1	Browse the web	Examinar la web
2	Browse the web	Examinar la web
3	Browse the web	Navegar por la web
4	Browse the web	Examinar la web
5	Browse the web	Surfear la web

If you edit the first translated string and set it to "Navegar por la web", strings 2 and 4 will also change their translation to "Navegar por la web", while string 5 will keep as "Surfear la web".

Search similar strings (experimental)

This query uses the Soundex indexing algorithm (available at <http://www.archives.gov/genealogy/census/soundex.html>) to find original strings with similar sounds and lists them.

Relink Labels & Keys

This option walks through the entire database redoing links between labels, commandkeys and accesskeys. For more information, see "Labels, commandkeys and accesskeys" in the User Guide.

Clear Fuzzy Flag

When you update a product (by running either **File | Update Product** for JAR-based products or **Edit | SCM Import directory** for SCM-based products), new and modified strings get their Fuzzy flag set. When you edit these strings using the Edit Phrase dialog the Fuzzy mark is removed, and you can do it so manually in the Chrome View. This means that all strings with Fuzzy flag are pending or need a review.

However, if by any reason a lot of your strings have the Fuzzy flag and you still feel that they are correct and need no review, you can run this option and **ALL** strings will get their Fuzzy flag removed. You will be asked before proceeding, just in case you mistakenly choose this option.

View Fuzzy

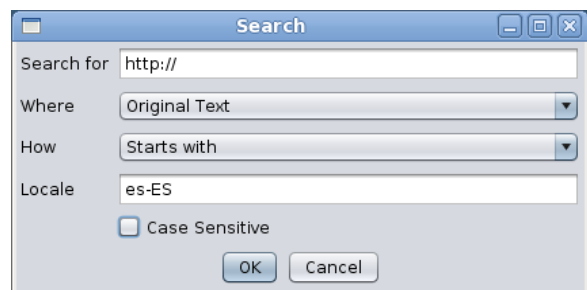
Surprisingly, this query gives you a list with all strings having their Fuzzy flag set. This may be used along with Autotranslate feature (see below).

View Binary

This gives you a list with all binary (ie.: images) strings.

Search

This option lets you input the string you're about to search, in which field look for, with which kind of match, the locale (relevant only if you're searching in the Translated text or the Comment field), and if it should care about the characters case. You will get a list with the matching strings, allowing you to edit translations in the same way as Chrome View.



Advanced Search

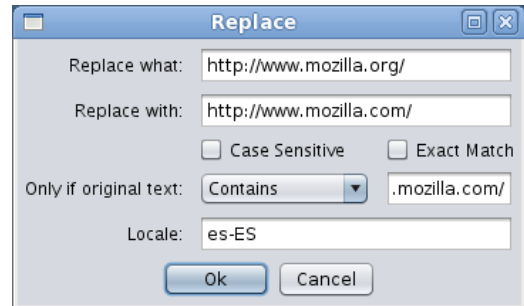
This option presents a less verbose dialog of the normal Search one, but allowing you specify up to three criteria that may be treated as compound



(AND) or alternative (OR) to match strings. Besides that, it behaves in the same way that regular search dialog.

Replace

This option allows you to do a replace in the translation. Exact match forces the “Replace what” string to be the whole content of the string, instead of being just a part of it. You may also add a condition based on the original text contents. You'll get a list with the replaced strings.



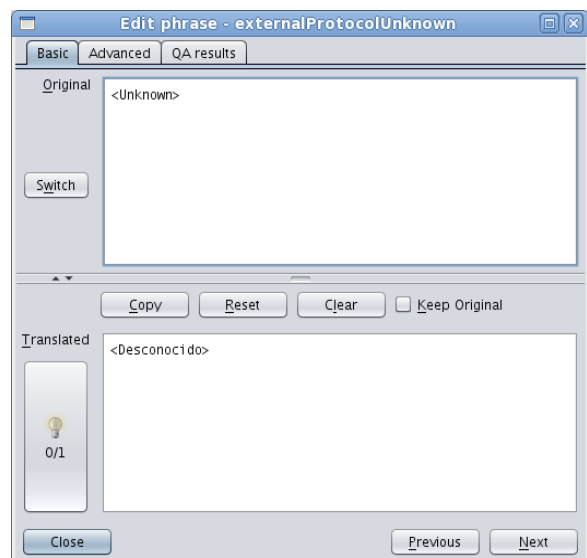
Edit Selected Phrase

This option does the same than pressing the button “Edit Selected Phrase” in the tool bar of Chrome View and query results. The Edit Phrase dialog is shown, which has the following elements (note that the key/entity name is displayed in the dialog title bar):

Basic Tab

This tab contains the most usual elements while translating, which are:

- **Original:** this field shows the en-US string value. If there is a localization note, changes to its content when the “Switch” button is pressed.
- **Translated:** here is where you can write or edit your translation. If there are translation suggestions, the light bulb button will be enabled, showing how many suggestions exist (more on this later).
- **Copy:** pressing this button copies the original value to the Translated field.
- **Reset:** pressing this button reverts to the value “Translated” field had before entering in the string being edited.
- **Clear:** pressing this button empties the translation.
- **Keep Original:** checking this will set the Keep Original flag.



Note: if you need to type an ellipsis (...) in the Translated textarea (ie., while translating), just press [Alt] + [.] and you will get it!

Note: if the translation suggestions button (the *light bulb button*) is enabled, you can simply press [Alt] + [s] to cycle over the entire suggestion list.




Translation suggestions

Translation suggestions appear when the original string you are translating is similar or equal to others existing anywhere in the glossary and having either a non-empty translation or a Keep Original flag set.

In both cases, the light bulb button will be enabled, and you can get translation suggestions in two different ways:

1. By pressing [Alt]+[S]. Translated area content will be replaced with translation suggestions, cycling over the entire list (the list doesn't filter out duplicates). As the list is traversed, the label under the light bulb tells you which translation suggestion has been applied last time (for instance: 1/3, 2/3, 3/3, and back to 1/3). If the label reads "0/3", it means that no translation suggestion has been retrieved yet.
2. By clicking in the light bulb button, a list will appear with all available translation suggestions (again, the list may contain duplicated values). You then click the one you wish and press OK, or Cancel to dismiss the window without applying any item.

As you are cycling through the suggestion list using the [Alt]+[S] keyboard shortcut, the light bulb color will tell you how good is the suggestion:

-  The original string used to provide this match is exactly equal to the one you are translating right now, and it is the only one in the glossary with that value.
-  The original string used to provide this match is exactly equal to the one you are translating right now, but there are several equal strings in the glossary.
-  The original string used to provide this match is similar, but not equal, to the one you are translating right now, so you may have to slightly edit the suggested translation.

Advanced Tab

This tab has mostly reference information about the string: product, components and file where it belongs, name of the entity/key, and so. If the string has an associated commandkey and/or accesskey, you will be able to set it here. You can also add a comment to the translation. Last, but not least, the Status field provides information about the translation status of the string, and can have these values:

- **Empty / Untranslated:** for non-empty strings with no translation (with or without Keep Original flag).
- **Modified:** for en-US strings that have been modified since last time they were translated.
- **Copied:** for en-US strings that have been auto-translated based on a 100% match with only one translation.
- **Proposed:** for en-US strings that have been auto-translated based on a 100% match with the most repeated translation among two or more.
- **Approximated:** for en-US strings that have been auto-translated based on a non-100% match with the best possible coincidence.
- **Translated:** for strings manually translated (ie., the user has edited the translation either in Chrome View or in the Edit Phrase Dialog).

The different Status values are meant to be used by MozillaTranslator Auto-translate feature, and can be manually set to whatever value at localizer's will. Ultimately, the localizer will want to have all (non-Keep Original flagged) strings set to Translated as a way to mark them as translated and reviewed.

QA results Tab

The QA results tab is useful after running a QA query or updating a product. In the first case it will have descriptive information about the reasons why a string has been selected in the query. In the second case (when updating), it will save the previous value for original strings that have changed.

Auto-Translate

This feature will look for untranslated strings, with an untranslated string defined as:

- one with no translation and not being marked as Keep Original.
- one with translation, whose Translation Status field is marked as **Untranslated**.

MT will take all strings having one of the above conditions and will do the following:

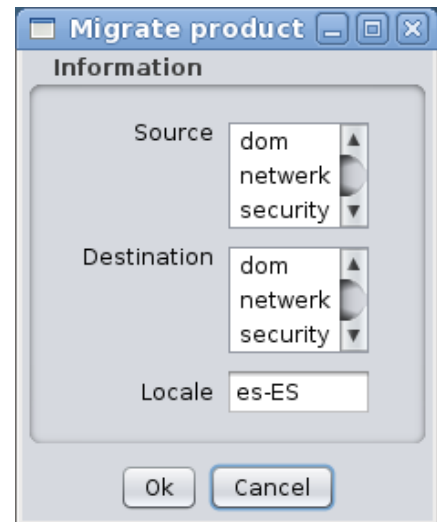
- If the original string is a commandkey, it will set its Keep Original flag. Commandkeys should not be translated most of the time.
- If the original string is an accesskey, it will set its Translation Status field to Untranslated.
- If the original string has a "DO_NOT_TRANSLATE" L10n note, it will set its Translation Status field to Untranslated.
- In the rest of cases, it will look for the best possible translation suggestion and fill it, setting the Translation Status field to:
 - Copied when there was just one 100% match suggestion candidate in the glossary.
 - Proposed when there was two or more 100% match suggestion candidates in the glossary.
 - Approximated when the chosen suggestion has a <100% match (the minimum match value can be set in MT preferences).

You will be then presented with a list of changed strings. Keep in mind that this features **DOES NOT** remove the Fuzzy flag.

Advanced Menu

Migrate Product

This option can be used to migrate a product to another similar one. It was mostly designed for quickly getting a Netscape 6.0/7.x translation starting from a Mozilla Suite 0.9/1.0 one. What this option does is walk through the source product components, files and keys and try to match each one with exactly the same component, file and key in the destination product. If it exists and has the same original text, then the translation will be copied, marking the string as Fuzzy.



Assign License

When you select this option, you will get a table view with Product, File and License columns. In this table, you may manually write the path in your file system to a file containing a license header valid for each file.

Nowadays, this option is not useful unless you plan to translate a product without license headers in the localizable files, as MT parses and cares about them (and it is a break of license to not respect and copy the license header in the original en-US file to the translated file, as this one is a derivative work from the en-US file).

Help Menu

About MozillaTranslator

This is just an About dialog with a funny talking dinosaur. :-) Since MT v5.25, it performs a `System.gc()` call, which forces Java Virtual Machine to look for unused memory and release it. It also gives some figures about memory usage.

Keyboard shortcut reference

Main menu options accelerator keys

These are the accelerator keys used for main menu options (invoked with [Alt]+[key]):

[F] -> File
[I] -> Import
[X] -> Export
[E] -> Edit
[A] -> Advanced
[H] -> Help

Common options shortcuts

These are the accelerator keys used for common menu options (invoked with [Ctrl]+[key], or [Ctrl]+[Shift], as noted):

Reserved for common use:

[C] -> Copy
[X] -> Cut
[V] -> Paste
[A] -> Select All
[Z] -> Undo (in edits)
[Y] -> Redo (in edits)

[M] File / Manage Products
[U] File / Update Product
[S] File / Save Glossary
[T] File / Save Settings
[Q] File / Quit

[O] Import / File

[L] Export / File
[J] Export / JAR
[I] Export / XPI

[H] Edit / Chrome View
[Shift]+[A] Edit / QA Queries / All above at once
[F] Edit / Search
[Shift]+[F] Edit / Advanced Search

[R] Edit / Replace
[E] Edit / Edit phrase

Special combinations

These are special keystroke combinations:

- [Alt]+[.] Type an ellipsis (...). Valid in Translated area of Edit Phrase Dialog and Translated cells enabled for editing in any table view.
- [Alt]+[s] Select next translation suggestion. Valid in Translated area of Edit Phrase Dialog when translation suggestions are enabled and available.