a folha

Boletim da língua portuguesa nas instituições europeias

http: //ec.europa.eu/translation/portuguese/magazine

English version of an article published in No. 55 — Autumn 2017

Free open source software — CAT Tools: OmegaT in DGT, its Wizard, Tagwipe and TeamBase (Part 2)

Maria José Bellino Machado; Hilário Leal Fontes — Portuguese Language Department
Thomas Cordonnier, Elio Fedele; Fons De Vuyst — Informatics Unit
Directorate-General for Translation — European Commission

[click here for Portuguese version]

Foreword

This is Part 2 of the article on Computer-Assisted Translation (CAT) applications which have been recently made available by the Directorate-General for Translation (DGT) of the European Commission as free open source software (FOSS).

[click here for Part 1]

Unlike DGT-OmegaT and Tagwipe which can be downloaded and used straightway, the two applications presented in this Part 2 require some work to be "usable" outside DGT, especially TeamBase:

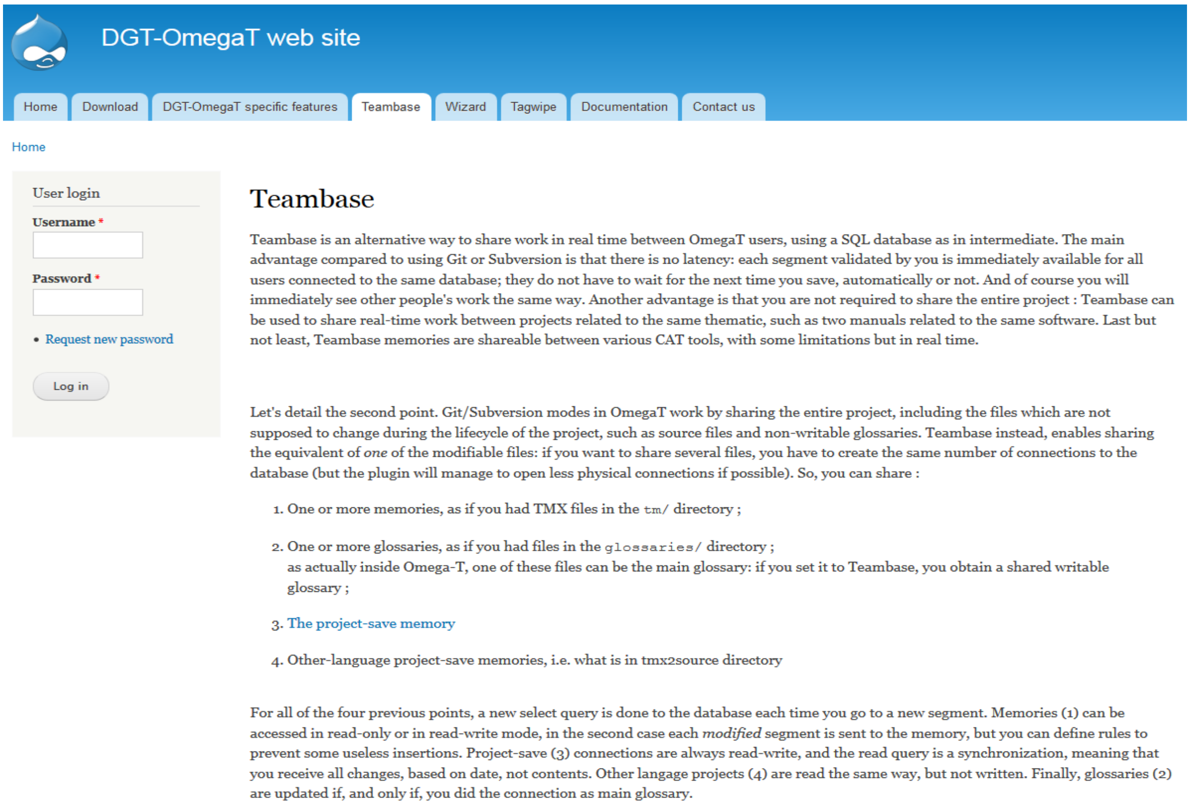
- **Teambase** (Developer: Thomas Cordonnier): An application initially developed privately by Thomas Cordonnier (Silvestris/Cyclotis project) which is being used in DGT and further customized to satisfy DGT's needs.
- **DGT-OmegaT Wizard** (Developer: Elio Fedele): a full in-house DGT application to integrate DGT-OmegaT in DGT's workflow, automating most of the project management operations.

The project is managed by Fons De Vuyst, Head of the Operational Support Sector in DGT's Informatics Unit.

The DGT-OmegaT Wizard can only be used in a Windows environment, while TeamBase can be used in any Java8-compliant platform (Windows, Linux, MacOS) and has plugins for OmegaT and SDL Trados Studio.

For both applications, the source code and the executable version are published, but as-is, without express or implied warranties, and come without support.

6. *Teambase*



Teambase

Teambase is an alternative way to share work in real time between OmegaT users, using a SQL database as in intermediate. The main advantage compared to using Git or Subversion is that there is no latency: each segment validated by you is immediately available for all users connected to the same database; they do not have to wait for the next time you save, automatically or not. And of course you will immediately see other people's work the same way. Another advantage is that you are not required to share the entire project : Teambase can be used to share real-time work between projects related to the same thematic, such as two manuals related to the same software. Last but not least, Teambase memories are shareable between various CAT tools, with some limitations but in real time.

Let's detail the second point. Git/Subversion modes in OmegaT work by sharing the entire project, including the files which are not supposed to change during the lifecycle of the project, such as source files and non-writable glossaries. Teambase instead, enables sharing the equivalent of *one* of the modifiable files: if you want to share several files, you have to create the same number of connections to the database (but the plugin will manage to open less physical connections if possible). So, you can share :

1. One or more memories, as if you had TMX files in the `tm/` directory ;
2. One or more glossaries, as if you had files in the `glossaries/` directory ;
   as actually inside Omega-T, one of these files can be the main glossary: if you set it to Teambase, you obtain a shared writable glossary ;
3. The project-save memory
4. Other-language project-save memories, i.e. what is in tmx2source directory

For all of the four previous points, a new select query is done to the database each time you go to a new segment. Memories (1) can be accessed in read-only or in read-write mode, in the second case each *modified* segment is sent to the memory, but you can define rules to prevent some useless insertions. Project-save (3) connections are always read-write, and the read query is a synchronization, meaning that you receive all changes, based on date, not contents. Other langage projects (4) are read the same way, but not written. Finally, glossaries (2) are updated if, and only if, you did the connection as main glossary.

Sometimes a translation job needs to be split among different translators either because it is too big or because it is urgent. When cutting a file in two or more parts and giving each part to a different translator, or assigning several related documents to different translators, it is not efficient and/or is more costly to ensure consistency across translations if the translators cannot access their colleagues' ongoing translation in real-time.

OmegaT already provides a solution for this: the team project feature, based on tools normally designed for software development, namely **Git** and **Subversion**. But this solution has some disadvantages, such as the incompatibility with other CAT tools.

So TeamBase was developed as an option to allow the sharing of project memories in real-time among translators using either DGT-OmegaT or SDL Trados Studio, something which is very important in DGT’s working environment. And TeamBase is being improved with user feedback!

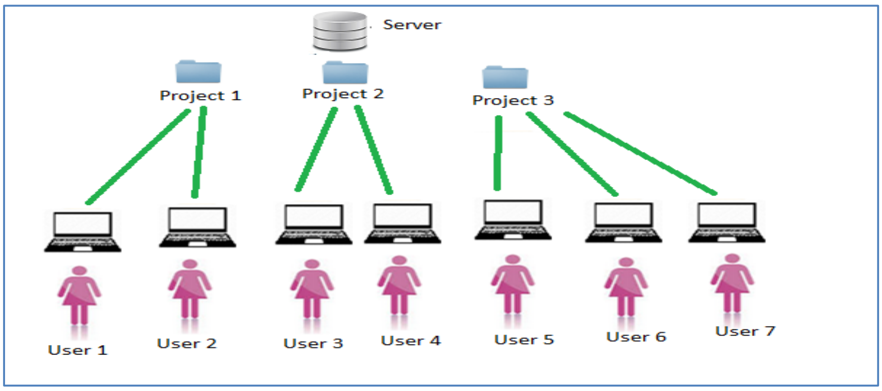
So that translators outside DGT can have an idea of what TeamBase is all about, to assess if it is of any interest to them/their work — and worth the trouble! — we present below a concise comparison, as objective as we can make it, between TeamBase and other options we know about[1].

	Shared directory (Windows share, cloud, ...)	Team projects (Git/Subversion)	TeamBase translation memory	TeamBase project memory
What can be shared	Directories: tm/, ...	All the project	Contents of one TMX file	Contents of project_save.tmx
What is not shared	project_save.tmx (unless manual work)		Dictionaries, project properties Source and target files	
			project_save.tmx	Translation memories
Synchronisation	Manual	At each save	At each segment validation	
External tools needed	None	A git or subversion **server**	A PostgreSQL **server**	
Integration inside OmegaT	N/A	Core OmegaT 2.6 or more	DGT-OmegaT (includes plugin) Or patched OmegaT 4.1+	
Compatibility with other CAT tools, namely SDL Trados Studio	Using TMX format	None (or read-only)	Good	Good (read-only)
When connection is lost	N/A (all is manual)	You work locally	You work locally You do not see distant work Your work is not sent to the server	A message says that connection is lost You work locally
When connection is back		Re-synchronisation	You see again segments from your colleagues Only segments modified after the re-connection are sent to the server	You receive all what has changed in the distant database Only segments modified after the re-connection are sent to the server

6.1. Client/server and TeamBase architecture

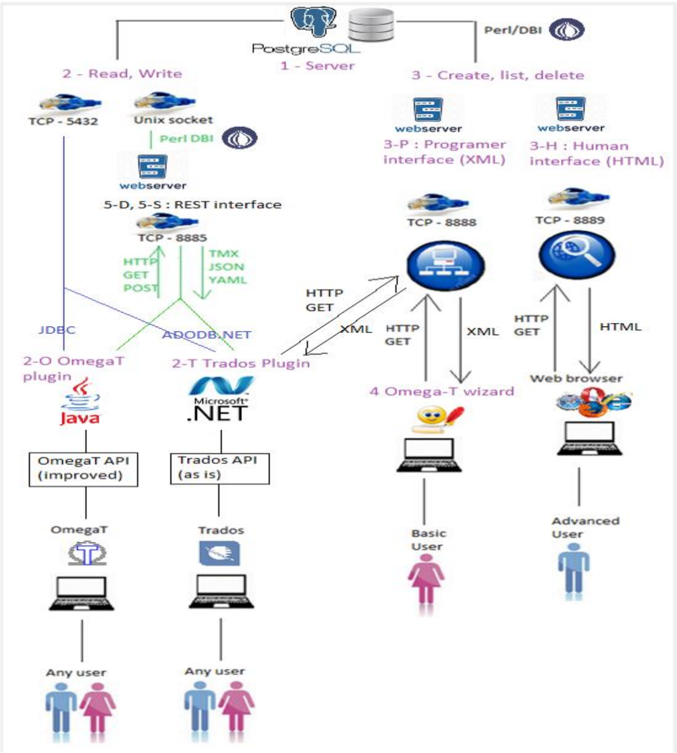
First, let's make a brief technical introduction to TeamBase client/server architecture. For detailed technical information on TeamBase and how to install and use it, see the DGT-OmegaT website[(2)].

Both OmegaT's team project and TeamBase are what is generally known as client-server architectures. In simple terms, that means that there is one computer called the "server", accessed only via the network (Internet or internal network), which stores the common data, and the users connect to the server via their own computer, called a "client". The server is generally unique and "serves" multiple projects and this way each project is shared by several users (see the illustration below).



Each user with DGT-OmegaT installed on a computer is a potential "client". Usually it is necessary to download separately the server, have/find a computer in a network to install it (usually, but not necessarily, a computer using a Unix distribution), install the programs and configure them depending on how the relevant user/organization manages its projects. Inside DGT we have one server, but it is only available internally and can only be shared within the DGT environment.

For use outside DGT, TeamBase can be installed in any server which fulfils its requirements.
Its installation is not as straightforward as for the other 3 applications presented in this article.
Just have a look at this screenshot to see its structure.
But don't be scared!
It can be done… with a little help if necessary!



Of course that, after being installed on a server, TeamBase can be used — and the settings defined — directly with all the features and options available.

In DGT, TeamBase memories are created via the DGT-OT Wizard making it much simpler and user-friendly than via the web-based creation interface… although not all the features and options will be available. Default settings are automatically defined without the user's intervention, in a background operation, when a connection to TeamBase is made.

Doing the same with the public version of the DGT-OT Wizard, once TeamBase is installed in the user's server, should work without problems if it does not need any authentication (it is just a matter of setting the correct URL in the configuration file). The use of TeamBase via the Wizard outside DGT is now under test (namely concerning its use with authenticated servers).

6.2. Creating a project to be shared

To create a project on the server it is necessary:

1 — To have TeamBase installed, of course, and its settings defined.
2 — To create an OmegaT project as usual.

> In this article we will, for simplicity's sake, use as an example a single-document project which is to be translated by 2 translators. But TeamBase can be as easily used for (large) multi-document projects.

For detailed information, see the DGT-OmegaT website.

6.3. TeamBase modes

Teambase enables sharing the equivalent of *one* of the modifiable files: several files can be shared and the same number of connections to the database can be created. So it is possible to share:

1. One or more memories, as if the translator had TMX files in the **\tm** directory;
2. One or more glossaries, as if the translator had files in the **\glossary** folder; the main glossary can work as a shared writable glossary;
3. The **project_save** memory;
4. Other-language **project_save** memories, i.e. what is in **\tmx2source** folder;

The first option is generally known as **Memory Mode**, option 2 as **Glossary Mode** and options 3 and 4 as **Project Mode**.

Let’s now see each mode in some detail.

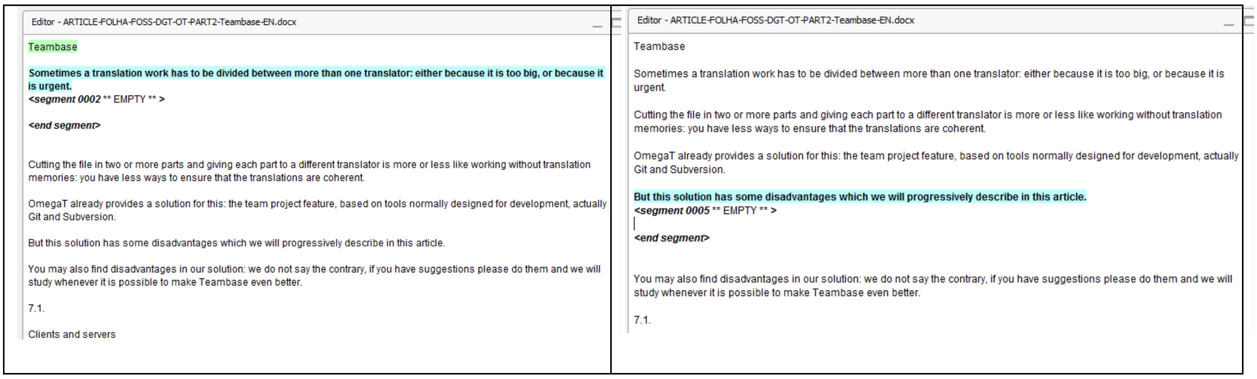
6.3.1. Project Mode

In this mode, TeamBase is used as a substitution of the ***project_save.tmx*** file: each time a translator validates the translation of a segment (even if s/he doesn’t save it, contrarily to OmegaT's project mode), the translation is sent to the database. In the other direction, other users will see the translated segment inserted in their project once they validate another segment.

To use TeamBase, all translators must have a copy of the project, including the file ***project_save.properties***. When they open the project in OmegaT, they will see it displayed as usual.

Let’s see what happens when Translator 1 opens segment 5 while Translator 2 has segment 1 opened in the **Editor**. Here we selected nearby segments only to make the behaviour immediately visible: in real usage, users will probably be in another part of the document.

Translator 1 translates segment 1 and validates it. At this moment, the translators do not see the same thing anymore:



Editor - ARTICLE-FOLHA-FOSS-DGT-OT-PART2-Teambase-EN.docx

Teambase

Sometimes a translation work has to be divided between more than one translator: either because it is too big, or because it is urgent.
<segment 0002 ** EMPTY ** **>**

<end segment>

Cutting the file in two or more parts and giving each part to a different translator is more or less like working without translation memories: you have less ways to ensure that the translations are coherent.

OmegaT already provides a solution for this: the team project feature, based on tools normally designed for development, actually Git and Subversion.

But this solution has some disadvantages which we will progressively describe in this article.

You may also find disadvantages in our solution: we do not say the contrary, if you have suggestions please do them and we will study whenever it is possible to make Teambase even better.

7.1.

Clients and servers

Editor - ARTICLE-FOLHA-FOSS-DGT-OT-PART2-Teambase-EN.docx

Teambase

Sometimes a translation work has to be divided between more than one translator: either because it is too big, or because it is urgent.

Cutting the file in two or more parts and giving each part to a different translator is more or less like working without translation memories: you have less ways to ensure that the translations are coherent.

OmegaT already provides a solution for this: the team project feature, based on tools normally designed for development, actually Git and Subversion.

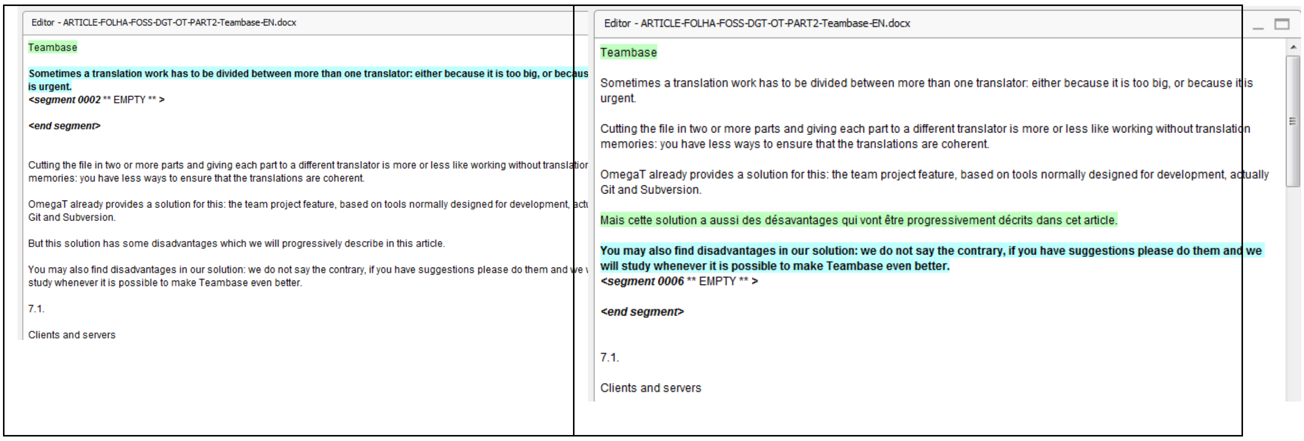
But this solution has some disadvantages which we will progressively describe in this article.
<segment 0005 ** EMPTY ** **>**

<end segment>

You may also find disadvantages in our solution: we do not say the contrary, if you have suggestions please do them and we will study whenever it is possible to make Teambase even better.

7.1.

At this point, synchronization has not yet been done for Translator 2: it will only be done when (s)he does something like opening another segment or validating a translation, as shown below.



Editor - ARTICLE-FOLHA-FOSS-DGT-OT-PART2-Teambase-EN.docx

Teambase

Sometimes a translation work has to be divided between more than one translator: either because it is too big, or becaus is urgent.
<segment 0002 ** EMPTY ** **>**

<end segment>

Cutting the file in two or more parts and giving each part to a different translator is more or less like working without translatior memories: you have less ways to ensure that the translations are coherent.

OmegaT already provides a solution for this: the team project feature, based on tools normally designed for development, actu Git and Subversion.

But this solution has some disadvantages which we will progressively describe in this article.

You may also find disadvantages in our solution: we do not say the contrary, if you have suggestions please do them and we v study whenever it is possible to make Teambase even better.

7.1.

Clients and servers

Editor - ARTICLE-FOLHA-FOSS-DGT-OT-PART2-Teambase-EN.docx

Teambase

Sometimes a translation work has to be divided between more than one translator: either because it is too big, or because it is urgent.

Cutting the file in two or more parts and giving each part to a different translator is more or less like working without translation memories: you have less ways to ensure that the translations are coherent.

OmegaT already provides a solution for this: the team project feature, based on tools normally designed for development, actually Git and Subversion.

Mais cette solution a aussi des désavantages qui vont être progressivement décrits dans cet article.

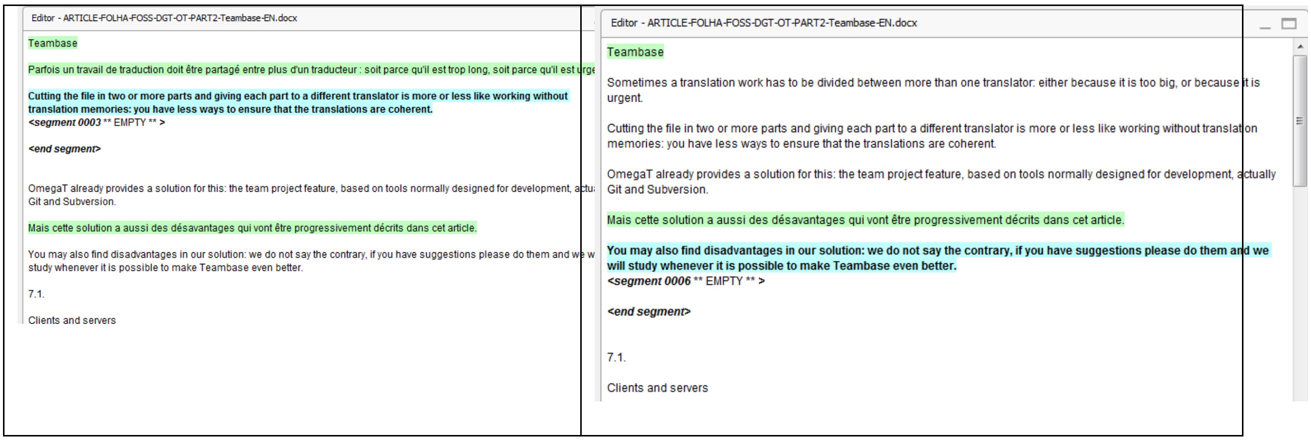
You may also find disadvantages in our solution: we do not say the contrary, if you have suggestions please do them and we will study whenever it is possible to make Teambase even better.
<segment 0006 ** EMPTY ** **>**

<end segment>

7.1.

Clients and servers

Now that Translator 2 has validated a segment, s/he also sees, on the top, the segment validated by Translator 1. In the other computer, nothing changed. But now let's see what happens when Translator 1 again validates a segment:



Editor - ARTICLE-FOLHA-FOSS-DGT-OT-PART2-Teambase-EN.docx

Teambase

Parfois un travail de traduction doit être partagé entre plus d'un traducteur : soit parce qu'il est trop long, soit parce qu'il est urge

Cutting the file in two or more parts and giving each part to a different translator is more or less like working without translation memories: you have less ways to ensure that the translations are coherent.
<segment 0003 ** EMPTY ** **>**

<end segment>

OmegaT already provides a solution for this: the team project feature, based on tools normally designed for development, actu: Git and Subversion.

Mais cette solution a aussi des désavantages qui vont être progressivement décrits dans cet article.

You may also find disadvantages in our solution: we do not say the contrary, if you have suggestions please do them and we w study whenever it is possible to make Teambase even better.

7.1.

Clients and servers

Editor - ARTICLE-FOLHA-FOSS-DGT-OT-PART2-Teambase-EN.docx

Teambase

Sometimes a translation work has to be divided between more than one translator: either because it is too big, or because it is urgent.

Cutting the file in two or more parts and giving each part to a different translator is more or less like working without translation memories: you have less ways to ensure that the translations are coherent.

OmegaT already provides a solution for this: the team project feature, based on tools normally designed for development, actually Git and Subversion.

Mais cette solution a aussi des désavantages qui vont être progressivement décrits dans cet article.

You may also find disadvantages in our solution: we do not say the contrary, if you have suggestions please do them and we will study whenever it is possible to make Teambase even better.
<segment 0006 ** EMPTY ** **>**

<end segment>

7.1.

Clients and servers

This time, nothing changes on the Translator 2's screen. But look at the Translator 1's screen, on the left: once he has validated the translation for segment 2, s/he can also see the translation of segment 5, which has been validated by the other translator.

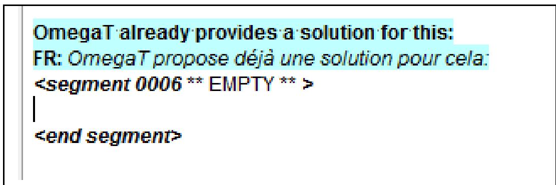
Besides this, working with Teambase in Project Mode is virtually the same as working alone, the only noticeable difference being that each translator receives all the other translators' translated segments as soon as they are validated (either with **Enter** or by leaving the newly translated or modified segment in any other way).

So translators sharing a project will work with OmegaT as usual, except that sometimes a new segment (for that translator) is displayed with a green background, something which in this case means that it has already been translated by a colleague sharing the project.

For translators who are familiar with OmegaT's "team projects", this should not be a surprise: in this mode as well, the translator also receives translations from other colleagues. The difference is in the frequency of the upgrade: for OmegaT's "team projects", this occurs on every save, so every 3 minutes, while for TeamBase it occurs immediately each time a segment is validated.

TeamBase and team projects have differences, but the common point is that they act **on request**: when a translation is submitted, it is sent to the server, but the server does not inform other clients that there is a new segment — the other clients will receive the new segment only when they ask the server again, and that occurs when they open a new segment — whether they added a new translation or not. A pull-based service, where the server informs all its clients that there is a new segment, would be more complicated to develop… and not really necessary!

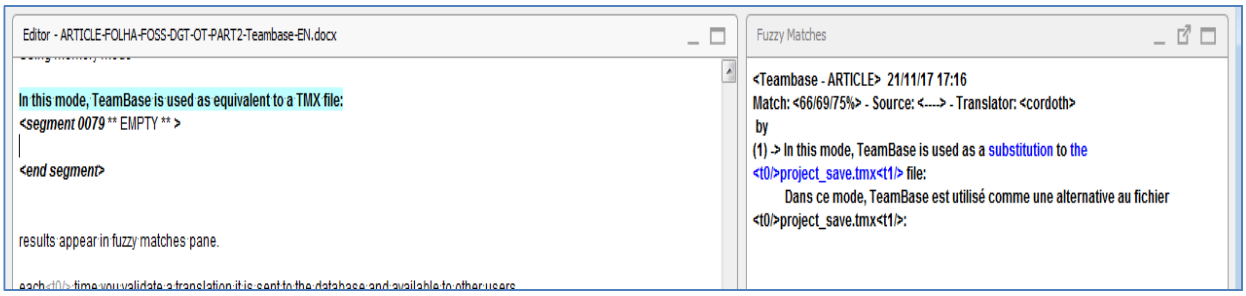
One last point about the Project Mode: until now we have used it as a substitute for ***project_save.tmx,*** but in the beginning of this section, we also said that it can be used as a substitute for **tmx2source.** If the **properties** file is placed in the **tmx2source** folder and named as, for example, FR.**properties**, then the result will appear in the **Editor** as if it came from the **FR.tmx** memory:

 OmegaT already provides a solution for this: FR: *OmegaT propose déjà une solution pour cela:* ***<segment 0006*** ** EMPTY ** **>** ***<end segment>***	Using a database instead of a TMX file makes sense only if the other-language translator is working simultaneously on that document: all segments validated by that other-language translator in OmegaT will be displayed, even if (s)he is still working on it.

6.3.2. Memory Mode

In this mode the translators are basically sharing a translation memory (like a TMX, except that it is outside each translator's computer and can be in **Read/Write** mode).

All features available for TMX files (including concordance if DGT-OmegaT version 3 is used) are valid for this memory, plus the fact that, in **Write** mode, it will also receive a copy of the segments that are being translated, making them immediately available for the translators connected to the same memory, as shown below.



In this mode, TeamBase is used as equivalent to a TMX file:
<segment 0079 ** EMPTY ** >
<end segment>

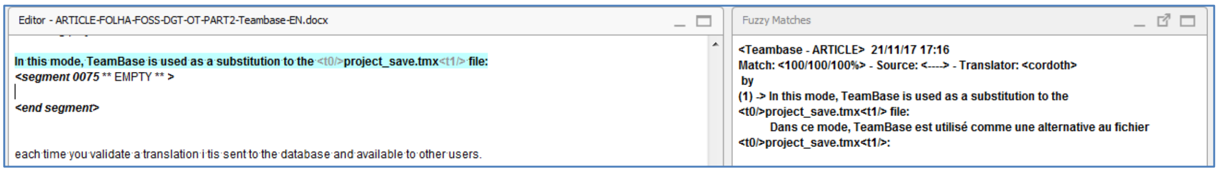
results appear in fuzzy matches pane.

each<t0/>time you validate a translation it is sent to the database and available to other users

<Teambase - ARTICLE> 21/11/17 17:16
Match: <66/69/75%> - Source: <----> - Translator: <cordoth>
by
(1) -> In this mode, TeamBase is used as a substitution to the <t0/>project_save.tmx<t1/> file:
Dans ce mode, TeamBase est utilisé comme une alternative au fichier <t0/>project_save.tmx<t1/>:

In this example, a segment was found in TeamBase which has a 66% match to the current one. So, it appears in the **Fuzzy Matches** pane, as if it came from a TMX file.

But what happens if there is a 100% match in TeamBase?



In this mode, TeamBase is used as a substitution to the <t0/>project_save.tmx<t1/> file:
<segment 0075 ** EMPTY ** **>**
<end segment>

each time you validate a translation i tis sent to the database and available to other users.

<Teambase - ARTICLE> 21/11/17 17:16
Match: <100/100/100%> - Source: <----> - Translator: <cordoth>
by
(1) -> In this mode, TeamBase is used as a substitution to the <t0/>project_save.tmx<t1/> file:
Dans ce mode, TeamBase est utilisé comme une alternative au fichier <t0/>project_save.tmx<t1/>:

The result is the same: a score of 100% is displayed, but that's all — translation memories, such as TMX files or TeamBase in Memory Mode, are only given as information, they are not part of the project memory. That is the difference in Project Mode: a segment translated by one translator is equivalent to a segment translated by another translator.

However, it may happen that, when a translator opens a segment, OmegaT automatically inserts something in it. In this case it means that the **Insert the best fuzzy match** is activated in the **Editor**

behaviour menu. This feature has nothing to do with TeamBase: whether the translator is in Project Mode, Memory Mode or not at all connected to TeamBase, (s)he may see the same behaviour if the segment comes from a TMX or even from a similar segment present in the document being translated.

In DGT-OmegaT, we decided to use a special colour (dark yellow background for fuzzy matches or dark green background for 100% matches) to make the distinction between **Insert best fuzzy match** and segments coming from the project memory (not to forget that, in Project Mode, TeamBase ***is*** the project memory). We hope that the colour makes it easier for translators to see the difference.

But what happens now if Translator 1 validates this segment?

In Project Mode, OmegaT would apply the same rule as for ***project_save.tmx***: either it is a default translation and it is replaced, or the translator creates an alternative translation, and unless the context (previous/next segments) is the same, both are kept by the server.

In Memory Mode, things are not so strict and there are several options. The translator can choose to:

- **Keep all entries**: each segment is saved in TeamBase, even if a segment with identical source already existed;
- **Keep only last one**: if a segment existed with the same source (strictly, including tags) then the new segment replaces the previous one;
- **Keep one entry per author**: if the translator submits a segment which (s)he has already translated, the server will replace it; but if it was associated to another author, then both are kept. This is the option generally preferred inside DGT and which is configured by default in the Wizard;
- **Keep one entry per context**: as in the Project Mode.

This must be chosen when the TeamBase memory is created and cannot be changed later. However, other memories can be created in the same server with different parameters. In all cases, the parameters defined at creation time will be kept throughout the life of each of the memories.

Memory Mode is useful when users are not really working on the same project, but on similar ones (two documents on the same subject, for example), as the Project Mode absolutely requires 100% matches.

On the other hand, if both translators have the same document in the source folder and decide that each translator has his/her own part to translate, then Memory Mode is not appropriate, due to the fact that auto-insertion is not done in non-visited segments and features such as "generate translated documents" will produce the other part in the original language.

Memory Mode is also useful if a translator working with DGT-OmegaT wants to share his/her work with others who are using SDL Trados Studio: due to differences in the way both tools store their context, their tags or their segmentation, there will be less 100% matches and more partial matches (even if the document to be translated is the same) compared to a full OmegaT setup in which all translators are working with DGT-OmegaT. So Project Mode cannot work in a heterogeneous environment. Concordance in this multi-tool setup will not be affected as long as tags are avoided in searches.

Memory Mode also supports the notion of Inheritance, as in object-oriented programming. For example, let's say that the translator created a memory named "Agriculture" to share it with all translators working on projects on this subject. But at a certain point, some translators thought that the theme was too general and created another memory named "Cereal".

With Inheritance, when a project is connected to "Cereal", all the segments are sent to this specific database, but translators connected to other children of "Agriculture", for example "Meat", will also see, if they activated Inheritance, all the segments from "Agriculture" and all children, without writing anything to them.

A general thing to remember is that, in TeamBase, more than one Memory Mode can be used in the same project simultaneously with one Teambase in the Project Mode. The only thing that must be avoided is to connect simultaneously in Project Mode and Memory Mode to the same TeamBase memory.

6.3.3. Glossary Mode

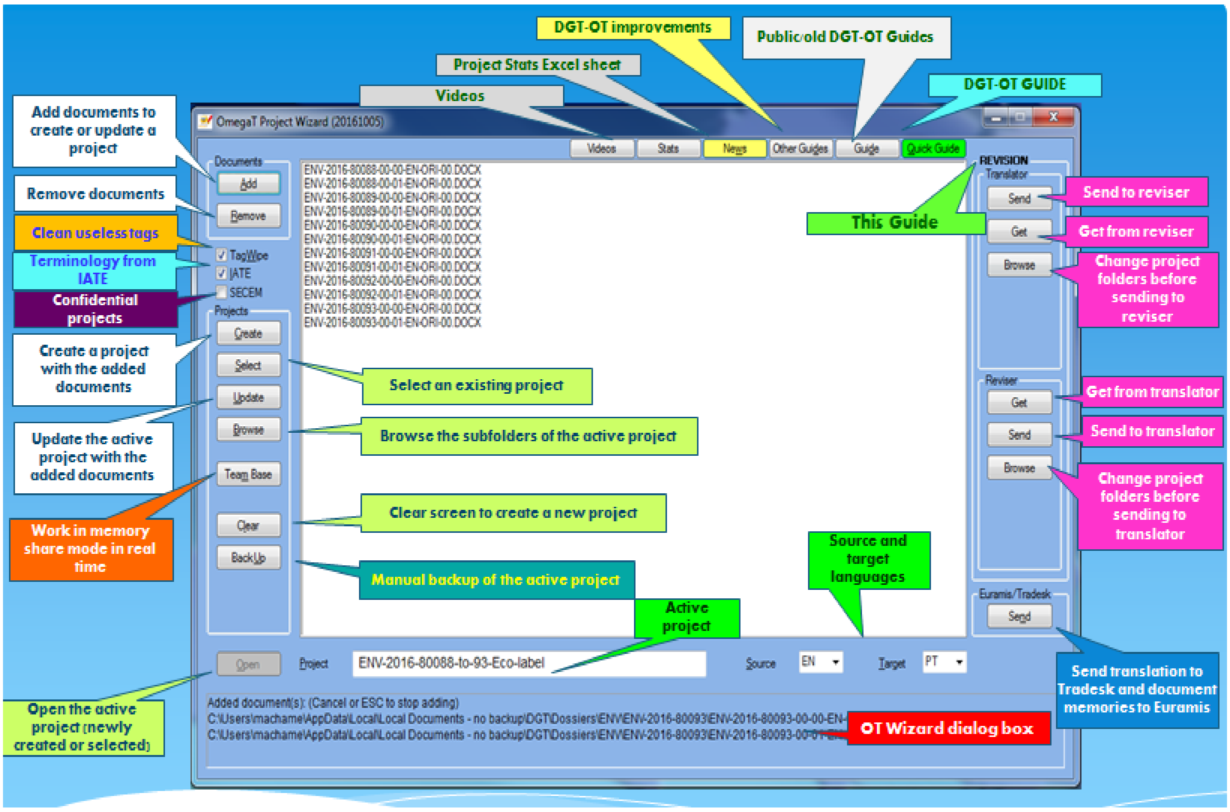
The Glossary Mode makes it possible to host in the server a shared version of the "writable glossary" provided by OmegaT: instead of it being in a file in the user computer, it is in the server and each time a new entry is added it is immediately available to other users.

Glossary Mode has support for Inheritance, like for Memory Mode.

7. DGT-OmegaT Wizard

DGT has developed in-house a DGT-OmegaT Project Wizard (DGT-OT Wizard) to integrate DGT-OmegaT in its workflow.

In this article, we present the workflow as used in DGT and we highlight some of the Wizard features.



OmegaT Project Wizard (20161005)

ENV-2016-80088-00-00-EN-ORI-00.DOCX
ENV-2016-80088-00-01-EN-ORI-00.DOCX
ENV-2016-80089-00-00-EN-ORI-00.DOCX
ENV-2016-80089-00-01-EN-ORI-00.DOCX
ENV-2016-80090-00-00-EN-ORI-00.DOCX
ENV-2016-80090-00-01-EN-ORI-00.DOCX
ENV-2016-80091-00-00-EN-ORI-00.DOCX
ENV-2016-80091-00-01-EN-ORI-00.DOCX
ENV-2016-80092-00-00-EN-ORI-00.DOCX
ENV-2016-80092-00-01-EN-ORI-00.DOCX
ENV-2016-80093-00-00-EN-ORI-00.DOCX
ENV-2016-80093-00-01-EN-ORI-00.DOCX

☑ TagWipe
☑ IATE
□ SECEM

Project: ENV-2016-80088-to-93-Eco-label
Source: EN
Target: PT

```
Added document(s): (Cancel or ESC to stop adding)
C:\Users\machame\AppData\Local\Local Documents - no backup\DGT\Dossiers\ENV\ENV-2016-80093\ENV-2016-80093-00-00-EN-
C:\Users\machame\AppData\Local\Local Documents - no backup\DGT\Dossiers\ENV\ENV-2016-80093\ENV-2016-80093-00-01-EN-
```

DGT-OT improvements
Public/old DGT-OT Guides
Project Stats Excel sheet
DGT-OT GUIDE
Videos
Videos
Stats
News
Other Guides
Guide
Quick Guide
Add documents to create or update a project
Remove documents
Clean useless tags
Terminology from IATE
Confidential projects
Create a project with the added documents
Update the active project with the added documents
Work in memory share mode in real time
Open the active project (newly created or selected)
Documents
Add
Remove
Projects
Create
Select
Update
Browse
Team Base
Clear
BackUp
Open
This Guide
Select an existing project
Browse the subfolders of the active project
Clear screen to create a new project
Manual backup of the active project
Active project
Source and target languages
REVISION
Translator
Send
Get
Browse
Reviser
Get
Send
Browse
Euramis/Tradesk
Send
Send to reviser
Get from reviser
Change project folders before sending to reviser
Get from translator
Send to translator
Change project folders before sending to translator
Send translation to Tradesk and document memories to Euramis
OT Wizard dialog box

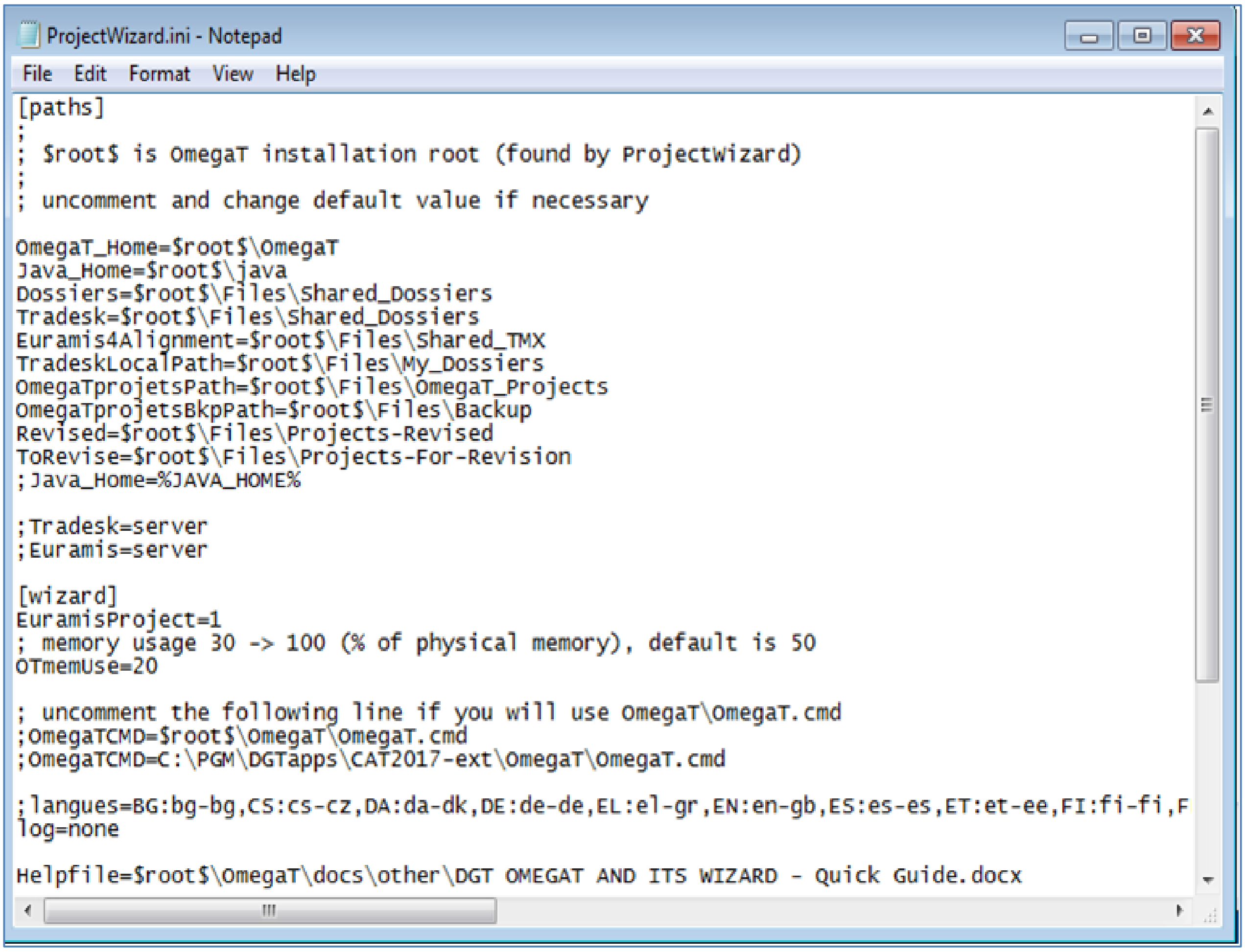
7.1. DGT-OmegaT Wizard for use outside DGT

The DGT-OT Wizard is made available as free open-source software both as binary (for Windows) and as source code. The binary includes DGT-OmegaT (standard OmegaT version 3.6.0.7 + DGT Extensions 3.0 beta, Update 8) so that both can be installed in a single operation.

Although it was developed to serve DGT needs in its particular context, it might also be useful for users outside DGT as OmegaT has only basic project management features. However, it will require some adaptation and not all the features will be available.

See the Section on the DGT-OT Wizard in the DGT-OmegaT website for detailed information on how to install and use the DGT-OT Wizard.

The main point is that, for use outside DGT, a **setup.ini** file has been added in which users can define the paths (either locally or on a server) in order to be able to use the Wizard — for most operations — in a different working environment. When the **setup.ini** file is saved (and the **setup.cmd** script is run), the Wizard will create automatically the folder structure (see Section 7.4).



```
[paths]
;
; $root$ is OmegaT installation root (found by ProjectWizard)
;
; uncomment and change default value if necessary

OmegaT_Home=$root$\OmegaT
Java_Home=$root$\java
Dossiers=$root$\Files\Shared_Dossiers
Tradesk=$root$\Files\Shared_Dossiers
Euramis4Alignment=$root$\Files\Shared_TMX
TradeskLocalPath=$root$\Files\My_Dossiers
OmegaTprojetsPath=$root$\Files\OmegaT_Projects
OmegaTprojetsBkpPath=$root$\Files\Backup
Revised=$root$\Files\Projects-Revised
ToRevise=$root$\Files\Projects-For-Revision
;Java_Home=%JAVA_HOME%

;Tradesk=server
;Euramis=server

[wizard]
EuramisProject=1
; memory usage 30 -> 100 (% of physical memory), default is 50
OTmemUse=20

; uncomment the following line if you will use OmegaT\OmegaT.cmd
;OmegaTCMD=$root$\OmegaT\OmegaT.cmd
;OmegaTCMD=C:\PGM\DGTapps\CAT2017-ext\OmegaT\OmegaT.cmd

;langues=BG:bg-bg,CS:cs-cz,DA:da-dk,DE:de-de,EL:el-gr,EN:en-gb,ES:es-es,ET:et-ee,FI:fi-fi,F
log=none

Helpfile=$root$\OmegaT\docs\other\DGT OMEGAT AND ITS WIZARD - Quick Guide.docx
```

Provided the user has the necessary IT skills, the Wizard source code can be changed/adapted to another workflow. However, if the above structure is created and the other necessary requirements are met, there is no need to change the Wizard.

7.2. DGT workflow

The DGT translation workflow is based on:

1) **Original documents for translation** — stored in the **Dossiers** server (DGT translation repository) and managed via **Tradesk,** which is DGT document management application (in the FOSS version, the folder suggested is **Shared_Dossiers)**.
2) **Pre-processed translation memories in TMX format** available in the **Dossiers** server in the relevant dossier folder under the **\pret** subfolder to be used in the creation and update of translation projects (in the FOSS version the folder suggested is **Shared_TMX**):
   a) From **Euramis** (DGT repository where aligned versions of EU legislation and documents translated in DGT and other EU institutions in the last 20 years are stored (without any formatting):
      i) **Retrievals**: Extraction to a TMX file of segments previously translated which match (65-100%) the relevant original(s) for translation;
      ii) **Downloads** (TMX files of full reference documents/legislation or previous versions/translations with a relevant match);
      iii) **Extraction to a TMX file of Eur-Lex legislation titles**.
   b) From **MT@EC** (DGT's in-house machine translation system based on Moses): Machine Translation output for the relevant document(s).
3) **Projects for revision and revised**, also stored in the **Dossiers** server and managed via **Tradesk** (in the FOSS version the folder suggested is **Projects-for-Revision** and **Projects-Revised**)
4) **Ongoing and finalized translations** stored in the **Dossiers** server via **Tradesk** (in the FOSS version the folder suggested is **Shared_Dossiers** as in point 1).

7.3. Operations that can be performed via the DGT-OT Wizard

The DGT-OT Wizard makes the link between Tradesk — DGT's document management system — and DGT-OmegaT and allows the user to:

a) Create single or multi-document projects with all the reference/retrieval/machine translation memories and also an IATE extraction. If the documents are in DOCX format, Tagwipe will clean useless tags;
b) Update projects with new original documents/new versions of documents already there;
c) Update projects with new memories;
d) Share the project memory via TeamBase;
e) Send projects for revision and get revised projects;
f) Send (draft or finalized) translated documents to Tradesk;
g) Send finalized document memories to Euramis.

Furthermore, the Wizard:

a) Does automatic backups (every 30 minutes) of the active project to a server (the personal space of the translator in the H: drive) in a background operation, in case there is a computer crash (in the FOSS version the folder suggested is **Backup**);
b) Sends automatically a copy of the project memory (draft) every 30 minutes to Tradesk ***\pret*** folder(s) of the relevant dossier(s) so that it can be used by another translator/reviser if necessary;
c) Copies all the memories (TMX files with tags) sent to Euramis via the DGT-OT Wizard to the ***\Final*** folder of the relevant dossier(s) in Tradesk for later reuse, either by a translator working in DGT-OmegaT or by a translator using Studio;
d) Copies all the memories sent to Euramis (via the Wizard) to the **_PROJECT-MEMORIES** subfolder of the **OmegaT_Projects** folder to gather all the memories of finalized translations sent to Euramis in tmx files with tags.

Furthermore, the Wizard gathers information on OmegaT and DGT-OmegaT: in-house and standard OmegaT Guides, News (with a list of the changes/improvements) and a video (not available outside DGT).

Users outside DGT can use the Wizard for most of operations provided that the relevant paths in the **setup.ini file** (either locally or on a server) are set and the relevant files are copied to the relevant folders.

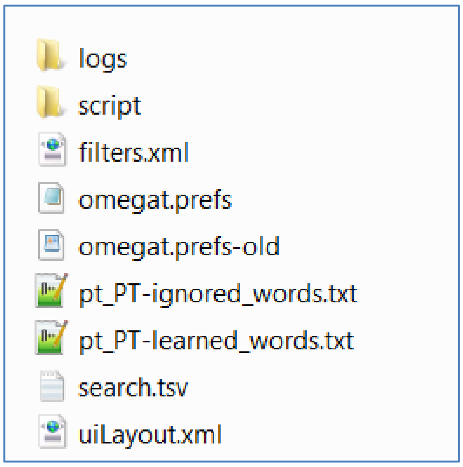
The internal DGT services — Dossier/Tradesk, Euramis, MT@EC — are replaced by simple file sharing.

7.4. Folders and project structure — Differences compared to OmegaT

7.4.1. OmegaT_Projects folder structure

When the DGT-OT Wizard is installed, it automatically creates:

a) A folder called **OmegaT_Projects** in which all the projects are created when using the Wizard.
b) A **CONFIG-PERSONAL** subfolder in the **OmegaT_Projects** folder where all the preferences, memorized searches and dictionaries with learned and ignored words are stored. This allows to easily change them.

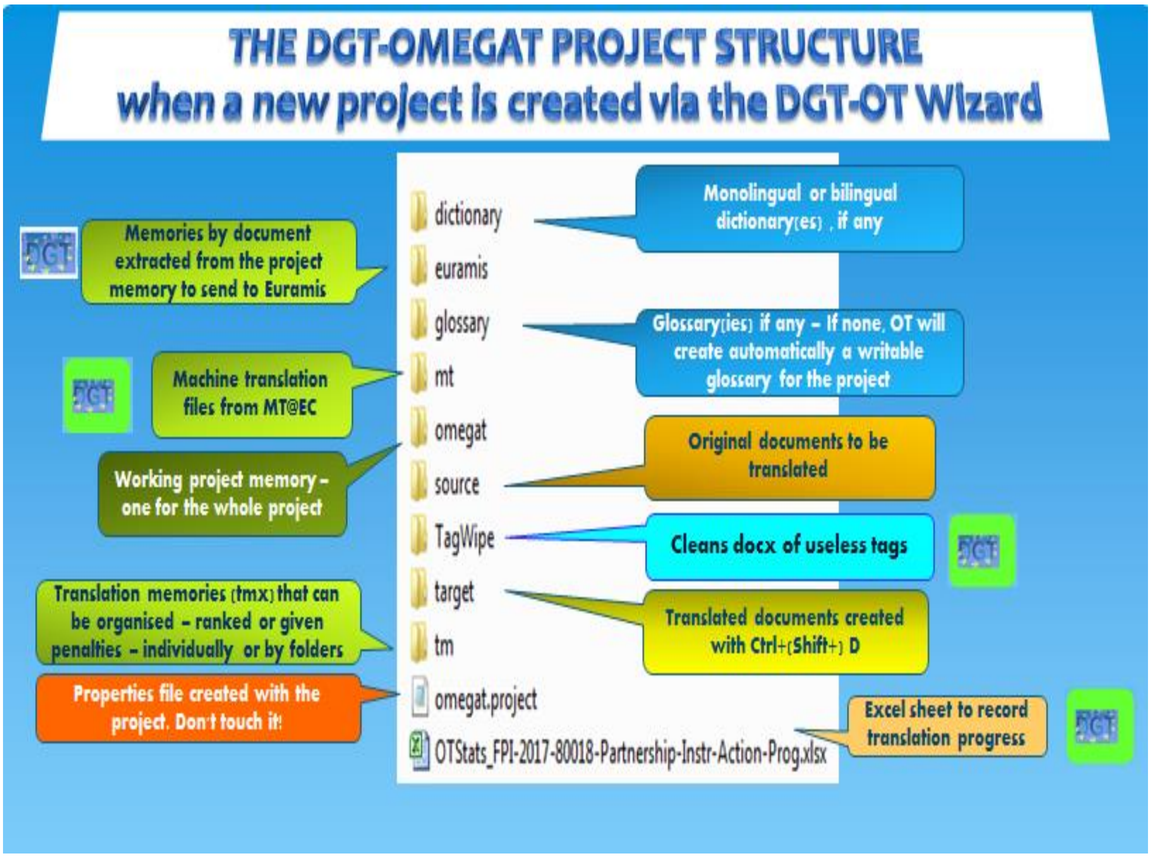


logs
script
filters.xml
omegat.prefs
omegat.prefs-old
pt_PT-ignored_words.txt
pt_PT-learned_words.txt
search.tsv
uiLayout.xml

c) A **_PROJECT-ARCHIVE** subfolder where, just by dragging and dropping them from the main **OmegaT_Projects** folder, finished projects can be stored so as not to have — over time — a long list of projects already finished in the main folder.
d) A **_PROJECT-MEMORIES** subfolder, to where the DGT-OT Wizard will automatically save a copy of the memories of the documents finalized with DGT-OmegaT sent to Euramis, thereby automatically gathering all the project memories (**with tags**) in a single folder.

7.4.2. Project folder structure

Compared to projects created with OmegaT, the projects created with the DGT-OT Wizard have the differences shown in the screenshot below:


THE DGT-OMEGAT PROJECT STRUCTURE
when a new project is created via the DGT-OT Wizard
dictionary
euramis
glossary
mt
omegat
source
TagWipe
target
tm
omegat.project
OTStats_FPI-2017-80018-Partnership-Instr-Action-Prog.xlsx
Monolingual or bilingual dictionary(es) , if any
Memories by document extracted from the project memory to send to Euramis
Glossary(ies) if any – If none, OT will create automatically a writable glossary for the project
Machine translation files from MT@EC
Working project memory – one for the whole project
Original documents to be translated
Cleans docx of useless tags
Translation memories (tmx) that can be organised – ranked or given penalties – individually or by folders
Translated documents created with Ctrl+(Shift+) D
Properties file created with the project. Don't touch it!
Excel sheet to record translation progress
DGT
DGT
DGT
DGT

a) An **\mt** subfolder is automatically created as DGT uses static MT provided in TMX files generated by its in-house Machine Translation service (MT@EC).

b) A **Tagwipe** subfolder is automatically created containing Tagwipe, making the application project-specific so that if, while translating/updating projects, Tagwipe is changed/improved, each project will not be affected and there will not be unduly untranslated segments.

c) In the **\omegat** subfolder, the **segmentation rules** of each project are also automatically made project-specific so that if, while translating/updating large projects, the segmentation rules are changed/improved, that particular project will not be affected (and there will be no unduly untranslated segments)

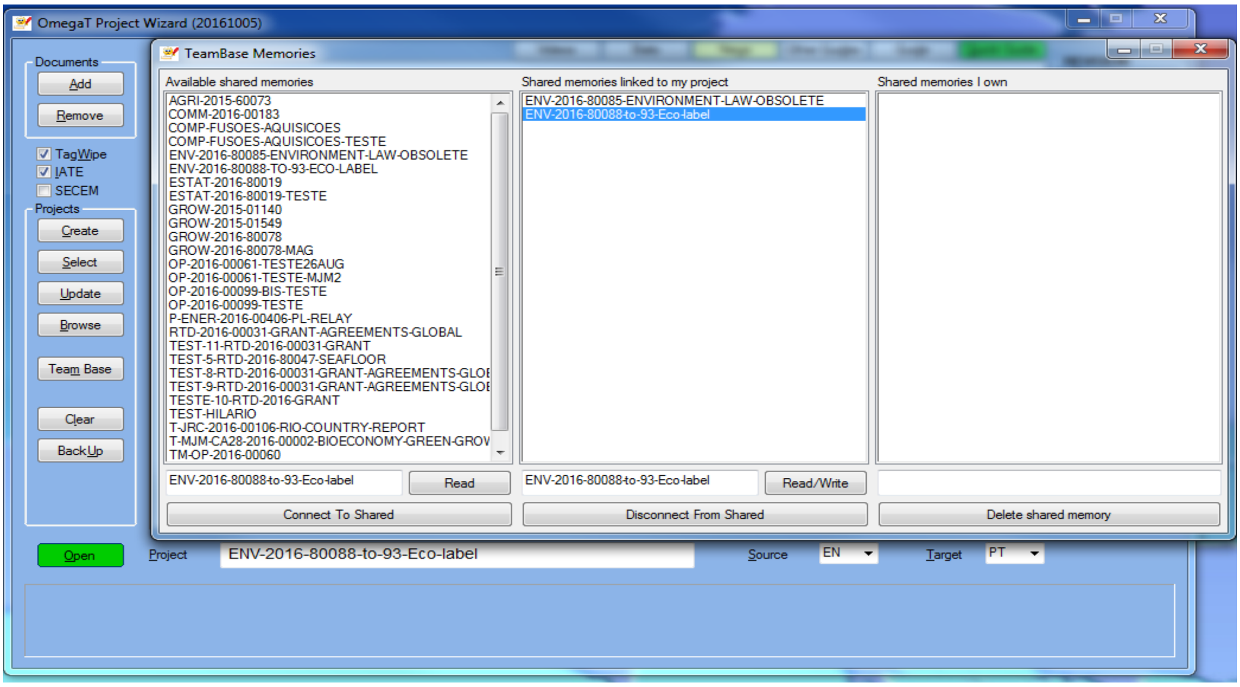
d) The **\tmx2source** and **penalty-50** subfolders are automatically created to be used for the **View Other Target Languages** feature (as explained in Part 1 of this article).

e) The **\euramis** subfolder is automatically created so that, when the individual translation memories of finished documents are generated to be sent to Euramis, it is in this folder that those files are stored. When they are sent to Euramis, the DGT-OT Wizard moves them to the **euramis\sent** subfolder. Each of these memories has all the segments (with tags) of a particular document without orphan segments and without notes.

7.5. TeamBase — Sharing memories in real time via the DGT-OT Wizard

TeamBase has already been explained in Section 6 and it can be used independently.

In this section we present briefly how it is used in DGT via the DGT-OT Wizard and how it can be used outside DGT after TeamBase is installed in the chosen server and the settings concerning TeamBase host path and password are defined.



☑ TagWipe
☑ IATE
□ SECEM

Available shared memories

AGRI-2015-60073
COMM-2016-00183
COMP-FUSOES-AQUISICOES
COMP-FUSOES-AQUISICOES-TESTE
ENV-2016-80085-ENVIRONMENT-LAW-OBSOLETE
ENV-2016-80088-TO-93-ECO-LABEL
ESTAT-2016-80019
ESTAT-2016-80019-TESTE
GROW-2015-01140
GROW-2015-01549
GROW-2016-80078
GROW-2016-80078-MAG
OP-2016-00061-TESTE26AUG
OP-2016-00061-TESTE-MJM2
OP-2016-00099-BIS-TESTE
OP-2016-00099-TESTE
P-ENER-2016-00406-PL-RELAY
RTD-2016-00031-GRANT-AGREEMENTS-GLOBAL
TEST-11-RTD-2016-00031-GRANT
TEST-5-RTD-2016-80047-SEAFLOOR
TEST-8-RTD-2016-00031-GRANT-AGREEMENTS-GLOE
TEST-9-RTD-2016-00031-GRANT-AGREEMENTS-GLOE
TESTE-10-RTD-2016-GRANT
TEST-HILARIO
T-JRC-2016-00106-RIO-COUNTRY-REPORT
T-MJM-CA28-2016-00002-BIOECONOMY-GREEN-GROV
TM-OP-2016-00060

ENV-2016-80088-to-93-Eco-label

Shared memories linked to my project

ENV-2016-80085-ENVIRONMENT-LAW-OBSOLETE
ENV-2016-80088-to-93-Eco-label

ENV-2016-80088-to-93-Eco-label

Shared memories I own

Project: ENV-2016-80088-to-93-Eco-label
Source: EN
Target: PT

The DGT-OT Wizard makes the use of TeamBase simple and user-friendly. To create a new TeamBase memory or to connect to one created by a colleague, it is just a matter of clicking on Teambase in the Wizard, either choosing a Teambase memory already created or accepting to create a new one (name is free) and selecting the desired mode (**Read** or **Read/Write**). The Wizard takes care of all the settings.

To disconnect from any of the TeamBase memories the user is connected to, it is just a matter of, in the Wizard, clicking on TeamBase and in the middle column — **Shared memories linked to my project** — highlight the memory to be disconnected from and clicking on **Disconnect From Shared**.

In the TeamBase window, in the left column are displayed the memories already created and which the translator can access in **Read** mode.

The translator can create a TeamBase memory, stay connected all the time or connect and disconnect from it at any time.

When creating a TeamBase memory the configuration settings are automatically selected without any action on the part of the user.

TeamBase, when used via the Wizard, uses the Memory Mode only — i.e. a TeamBase (TMX) memory which is automatically created on a server and which receives a copy of the newly translated or changed segments of all the translators connected to it in **Read/Write** mode (sending segments and receiving segments from all the translators linked to that particular TeamBase memory). In **Read** mode the translator only receives segments from other translators.

To be noted that, for segments already translated, only the reopened and changed segments in DGT-OmegaT will be sent to TeamBase when the **Read/Write** mode is selected. This means also that pre-translated segments will not be sent to Teambase either.

When using TeamBase via the Wizard, the parameters are pre-defined and cannot be changed via the Wizard.

The following settings are automatically taken from the DGT-OmegaT project created via the Wizard:

a) Project name by default (but the TeamBase memory name can be changed by the user when first connecting to TeamBase)
b) Source and target languages
c) Database host
d) Port
e) Database
f) User
g) Password

The following settings are pre-selected and cannot be changed via the Wizard:

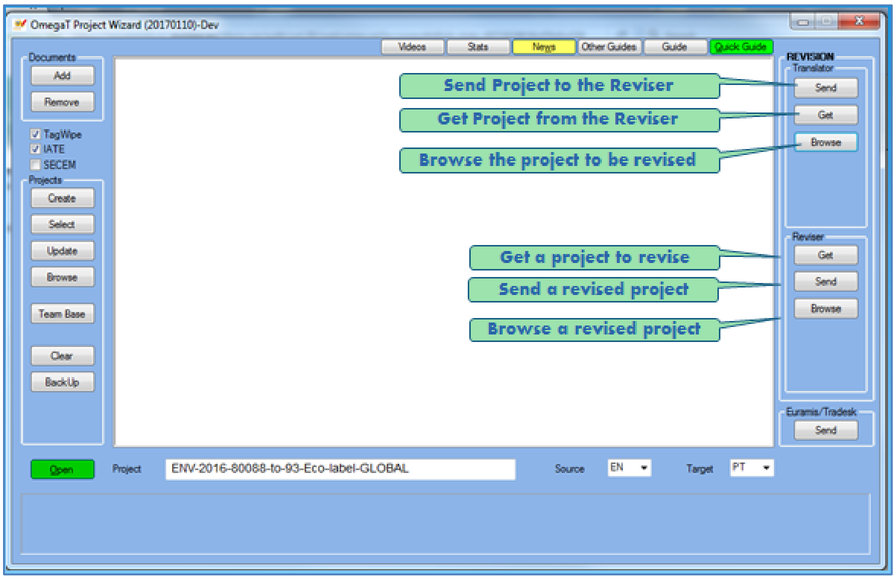
a) TeamBase mode: Memory Mode
b) Shared memory type: Free structure memory
c) Keeping segments with same source: One per author
d) Store properties

These TeamBase features cannot be used via the Wizard:

a) Project Mode
b) Shared glossaries
c) Inheritance

7.6. Revision workflow

The revision workflow for DGT-OmegaT users (both on translation and revision sides) has been automated via the DGT-OT Wizard.



Project: ENV-2016-80088-to-93-Eco-label-GLOBAL
Source: EN
Target: PT

The DGT-OT Wizard can be used by:

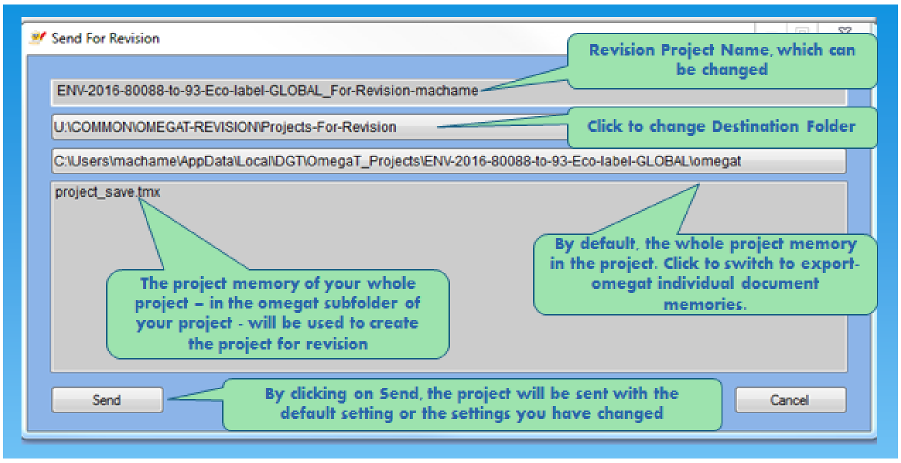
a) The **translator** to easily **Send** to the reviser the whole project for revision or, if necessary in case of multi-document projects, just a part of the project documents;
b) The **translator** to also **Browse** the project for revision — after sending it — in order to reorganise folders, for example, by deleting tmx files in the **\tm** subfolder of the project which will be of no interest to the reviser;

c) The **reviser** to **Get** the project for revision and copy it to his/her computer and afterwards to **Send** it back to the translator for finalization (if the translator has the last word);
d) The **reviser** to also **Browse** the revised project and reorganise it if s/he so wishes;
e) The **translator** to **Get** the revised project to finalize it.

Here we highlight only the sending of a project for revision as it is the step in the process which requires changes in the project structure/files. The other operations only require copying and renaming the project.

The default, in the **Send for Revision** window, is to copy the project to a default server location and the project is automatically reorganised by copying the project memory (***project_save.tmx*** in the \omegat subfolder) to the **tm\auto\draft** subfolder of the project for revision so that the reviser — even if s/he changes a segment — can always see, in the **Fuzzy Matches** pane, the segment as translated by the translator with track-changes displayed.

However, in the window that opens, the translator can choose to change any of the fields, just by clicking on any of them as shown in the screenshot below.



ENV-2016-80088-to-93-Eco-label-GLOBAL_For-Revision-machame

U:\COMMON\OMEGAT-REVISION\Projects-For-Revision

C:\Users\machame\AppData\Local\DGT\OmegaT_Projects\ENV-2016-80088-to-93-Eco-label-GLOBAL\omegat

project_save.tmx

Revision Project Name, which can be changed

Click to change Destination Folder

The project memory of your whole project – in the omegat subfolder of your project - will be used to create the project for revision

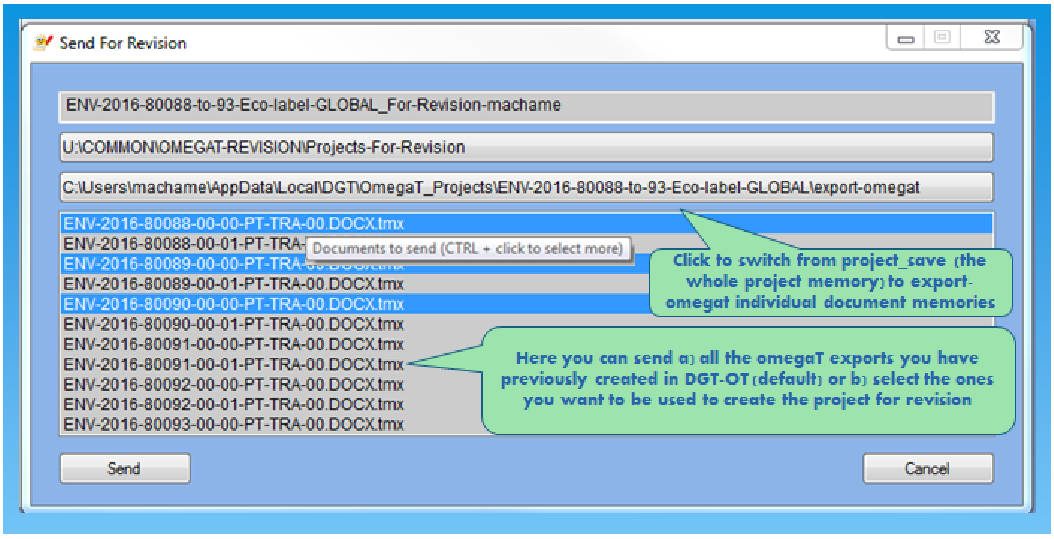
By default, the whole project memory in the project. Click to switch to export-omegat individual document memories.

By clicking on Send, the project will be sent with the default setting or the settings you have changed

But the translator may not want to send the whole project memory, either because:

a) In a multi-document project, only some of the documents are ready for revision; or
b) When there are many versions of one or more documents, there may be many ‘orphan’ segments (i.e., segments that were translated for previous versions but which no longer exist in the final document sent for revision and therefore were not carefully checked by the translator).

In that case, in DGT-OmegaT, the translator can run the command **Create OmegaT Export** of the document(s) to be sent for revision.



ENV-2016-80088-to-93-Eco-label-GLOBAL_For-Revision-machame

U:\COMMON\OMEGAT-REVISION\Projects-For-Revision

C:\Users\machame\AppData\Local\DGT\OmegaT_Projects\ENV-2016-80088-to-93-Eco-label-GLOBAL\export-omegat

ENV-2016-80088-00-00-PT-TRA-00.DOCX.tmx
ENV-2016-80088-00-01-PT-TRA-
ENV-2016-80089-00-00-PT-TRA-
ENV-2016-80089-00-01-PT-TRA-00.DOCX.tmx
ENV-2016-80090-00-00-PT-TRA-00.DOCX.tmx
ENV-2016-80090-00-01-PT-TRA-00.DOCX.tmx
ENV-2016-80091-00-00-PT-TRA-00.DOCX.tmx
ENV-2016-80091-00-01-PT-TRA-00.DOCX.tmx
ENV-2016-80092-00-00-PT-TRA-00.DOCX.tmx
ENV-2016-80092-00-01-PT-TRA-00.DOCX.tmx
ENV-2016-80093-00-00-PT-TRA-00.DOCX.tmx

Documents to send (CTRL + click to select more)

Click to switch from project_save (the whole project memory) to export-omegat individual document memories

Here you can send a) all the omegaT exports you have previously created in DGT-OT (default) or b) select the ones you want to be used to create the project for revision

8. *Acknowledgements*

We hope that these applications may be useful to users outside DGT as OmegaT has been useful to us.

And because people are important, we would like to thank:

- First of all **Didier Briel** and the **developers** of OmegaT for their remarkable work and for being so attentive to translators' needs… most of them being translators themselves!
- The **enthusiasts** that publish scripts, information, guides and videos that have helped a lot to understand how OmegaT works and to adapt it to our needs, namely **Vito Smolej**, **Susan Welsh** and **Marc Prior**, **Roman Mironov**, **Kos Ivantsov** and the many contributors in the OmegaT User Group.
- In **DGT**:
    - **Rytis Martikonis**, DGT Director-General, and **Dieter Rummel**, Head of DGT IT Unit, for being open-minded.
    - Colleagues from the Portuguese and Greek Language Departments: for their interest, feedback and questions, which allowed us to pinpoint what is more important for a (new) user in our working environment and especially: **Stratos Meintanopoulos (EL-LD)** and **Helena Lopes dos Santos, Anabela Pombo Rodrigues** and **Mário Vilar (PT-LD).**

Maria.Machado@ec.europa.eu
Thomas.Cordonnier@ext.ec.europa.eu
Hilario.Fontes@ec.europa.eu

[1] For a more in-depth comparison see Silvestris Laboratory, *Cyclotis inside OmegaT, compared to other solutions: What you share and don't share*, http://www.silvestris-lab.org/node/21#how-share.
[2] DGT-OmegaT web site, *Teambase*, http://185.13.37.79/?q=node/20.