**Temenos**

**User manual**

**Translation Tool**

**Translation of screen texts to any local language**

**Disclaimer:**

This Excel tool does not claim to be industrialised for common usage, there might be errors in the tool that are not handled. The Visual Basic for Applications (VBA) that is used for the tool is easy to read, and any T24 developer should be able to debug.

Support will only be provided if I there is availability, this tool does not come with any support guarantees from Temenos.

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Section(s) Amended |
| 0.1 | 20-07-2016 | Willem Gorter | Initial version |
| 0.2 | 19-09-2017 | Willem Gorter | Update for external use |
| 1.0 | 24-09-2017 | Willem Gorter | First Release version |
| 1.2 | 24-09-2020 | Willem Gorter | Update for OFS POST (R17 and above) |
| 1.2.2 | 12-10-2020 | Willem Gorter | Deployment of jars to jboss |
| 1.2.3 | 12-10-2020 | Willem Gorter | Environment picker |
| 1.3 | 13-10-2020 | Willem Gorter | Auto translate with Microsoft Azure translator |
| 1.3.1 | 22-03-2021 | Willem Gorter | Auto translate also with Google Translate API |

Document Sign Off

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Name | Company / Position |
|  |  |  |  |

Key Contacts

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Role | email | Phone |
| Willem Gorter | RSA | ewgorter@temenos.com | +45 24343891 |
|  |  |  |  |

Table of Contents

[1 Purpose 4](#_Toc55809698)

[2 Technical background 4](#_Toc55809699)

[2.1 TAFJ until R17 (this paragraph is now out of date) 4](#_Toc55809700)

[2.2 TAFJ as of R17 4](#_Toc55809701)

[2.2.1 TAFJ Application User 5](#_Toc55809702)

[3 Common features 6](#_Toc55809703)

[3.1 General configuration 6](#_Toc55809704)

[3.1.1 environment 6](#_Toc55809705)

[3.1.2 T24 User 6](#_Toc55809706)

[3.1.3 Max 6](#_Toc55809707)

[3.1.4 Function 7](#_Toc55809708)

[4 Installation 7](#_Toc55809709)

[4.1 Transferring and compiling source code 7](#_Toc55809710)

[4.2 Deploying the routines to jBoss 8](#_Toc55809711)

[4.2.1 Find the compiled routines 8](#_Toc55809712)

[4.2.2 Rename the jar, and copy it to the application server 8](#_Toc55809713)

[4.2.3 Add the new jar to the module list 9](#_Toc55809714)

[4.2.4 Subsequent updates 9](#_Toc55809715)

[4.3 Setting T24 environment variables 9](#_Toc55809716)

[4.3.1 Creating a list of environments 9](#_Toc55809717)

[4.3.2 Choosing the environment for execution 9](#_Toc55809718)

[4.4 Installing the T24 records for STANDARD.SELECTION and ENQUIRY 10](#_Toc55809719)

[4.5 Setting up the Microsoft Translator Service 11](#_Toc55809720)

[5 Usage 12](#_Toc55809721)

[5.1 Extraction 12](#_Toc55809722)

[5.1.1 Types of requests 13](#_Toc55809723)

[5.1.2 Launching a request 14](#_Toc55809724)

[5.2 Copy the translatable elements 14](#_Toc55809725)

[5.3 Correct English text 15](#_Toc55809726)

[5.4 Upload corrected English text 15](#_Toc55809727)

[5.5 Translate to Local language 15](#_Toc55809728)

[5.5.1 Automatic Translation with Microsoft Translator 15](#_Toc55809729)

[5.6 Upload translations 16](#_Toc55809730)

[5.7 Test and debug 17](#_Toc55809731)

[6 Disclaimer 17](#_Toc55809732)

[7 Appendix 17](#_Toc55809733)

[7.1 Creating a TAFJ user on a local T24 installation 17](#_Toc55809734)

# Purpose

The purpose of this document is to provide guidelines for using the Excel workbook TranslationTool\_v1\_2.xlsm (or later). The Excel workbook is a tool for translating T24 screen text to any local language, including right-to-left languages.

# Technical background

## TAFJ until R17 (this paragraph is now out of date)

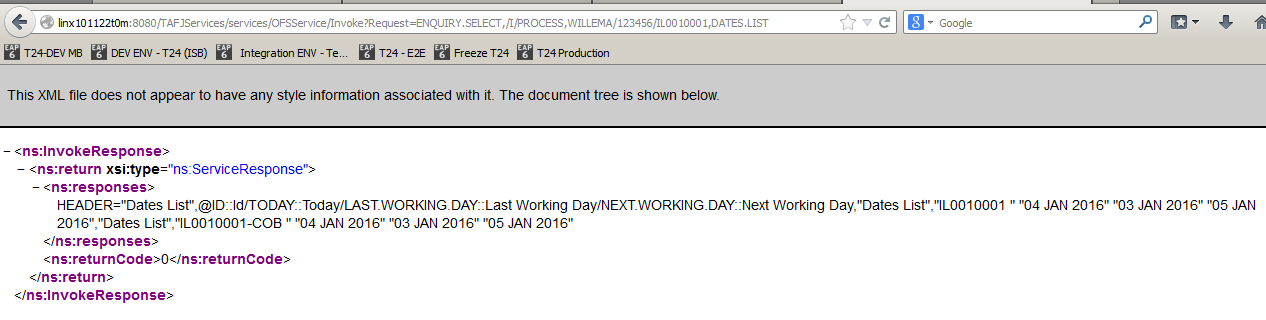
T24 browser (backoffice screens) uses OFS to communicate with T24 Server. In TAFJ, OFS can be run also directly as an HTTP-GET command from the address bar in a browser, if the address starts with:

<http://linx101122t0m:8080/TAFJServices/services/OFSService/Invoke?Request>=

(where the part starting with TAFJServices replaces the address of the Login screen starting with BrowserWeb). An example (it links to DEV server, from Leumi network you can Ctrl-click on it):

<http://linx101122t0m:8080/TAFJServices/services/OFSService/Invoke?Request=ENQUIRY.SELECT,/I/PROCESS,WILLEMA/123456/IL0010001,DATES.LIST>

will return, when pasted in Internet Explorer or Firefox:



The excel tool uses this mechanism to execute the actions. It will execute an HTTP-GET request, and read the response. The VBA code for this is:



To read the response, it will load the ResponseText property of the request, and parse it to find the OFS response. You can find the code in the Excel tool when editing macro's, in the routine LaunchOFSCommandTAFJ in module1.

## TAFJ as of R17

T24 browser (backoffice screens) uses OFS to communicate with T24 Server. In TAFJ, OFS can be run also directly as an HTTP-POST command, using Postman or any other API software, or through writing a program in any programming language, like Visual Basic for Applications, available in Excel.

The program should execute an HTTP request using (for example):

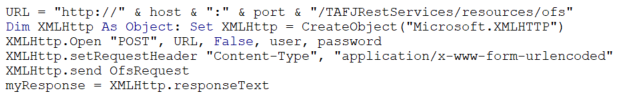
URL = <http://localhost:8089/TAFJRestServices/resources/ofs>  
Request = ofsRequest=ENQUIRY.SELECT,,INPUTT/123456,DATES.LIST  
Header = “Content-Type” = “application/x-www-form-urlencoded”

(where the URL part starting with TAFJServices replaces the address of the Login screen starting with BrowserWeb).

This will return, in the HTTP Response:

{"ofsRequest":"ENQUIRY.SELECT,,INPUTT/123456,DATES.LIST","ofsResponse":"HEADER=\"Dates List\",@ID::Id/TODAY::Today/LAST.WORKING.DAY::Last Working Day/NEXT.WORKING.DAY::Next Working Day,\"Dates List\",\"AU0010001 \"\t\"17 APR 2020\"\t\"16 APR 2020\"\t\"20 APR 2020\",\"Dates List\",\"AU0010001-COB \"\t\"17 APR 2020\"\t\"16 APR 2020\"\t\"20 APR 2020\",\"Dates List\",\"EU0010001 \"\t\"17 APR 2020\"\t\"16 APR 2020\"\t\"20 APR 2020\",\"Dates List\",\"EU0010001-COB \"\t\"17 APR 2020\"\t\"16 APR 2020\"\t\"20 APR 2020\",\"Dates List\",\"GB0010001 \"\t\"17 APR 2020\"\t\"16 APR 2020\"\t\"20 APR 2020\",\"Dates List\",\"GB0010001-COB \"\t\"17 APR 2020\"\t\"16 APR 2020\"\t\"20 APR 2020\",\"Dates List\",\"GB0010002 \"\t\"17 APR 2020\"\t\"16 APR 2020\"\t\"20 APR 2020\",\"Dates List\",\"GB0010002-COB \"\t\"17 APR 2020\"\t\"16 APR 2020\"\t\"20 APR 2020\",\"Dates List\",\"GB0010003 \"\t\"17 APR 2020\"\t\"16 APR 2020\"\t\"20 APR 2020\",\"Dates List\",\"GB0010003-COB \"\t\"17 APR 2020\"\t\"16 APR 2020\"\t\"20 APR 2020\",\"Dates List\",\"GB0010004 \"\t\"17 APR 2020\"\t\"16 APR 2020\"\t\"20 APR 2020\",\"Dates List\",\"GB0010004-COB \"\t\"17 APR 2020\"\t\"16 APR 2020\"\t\"20 APR 2020\",\"Dates List\",\"GB0010005 \"\t\"17 APR 2020\"\t\"16 APR 2020\"\t\"20 APR 2020\",\"Dates List\",\"GB0010005-COB \"\t\"17 APR 2020\"\t\"16 APR 2020\"\t\"20 APR 2020\",\"Dates List\",\"NL0010001 \"\t\"17 APR 2020\"\t\"16 APR 2020\"\t\"20 APR 2020\",\"Dates List\",\"NL0010001-COB \"\t\"17 APR 2020\"\t\"16 APR 2020\"\t\"20 APR 2020\",\"Dates List\",\"SG0010001 \"\t\"17 APR 2020\"\t\"16 APR 2020\"\t\"20 APR 2020\",\"Dates List\",\"SG0010001-COB \"\t\"17 APR 2020\"\t\"16 APR 2020\"\t\"20 APR 2020\",\"Dates List\",\"SG0020100 \"\t\"17 APR 2020\"\t\"16 APR 2020\"\t\"20 APR 2020\",\"Dates List\",\"SG0020100-COB \"\t\"17 APR 2020\"\t\"16 APR 2020\"\t\"20 APR 2020\",\"Dates List\",\"SG0020101 \"\t\"17 APR 2020\"\t\"16 APR 2020\"\t\"20 APR 2020\",\"Dates List\",\"SG0020101-COB \"\t\"17 APR 2020\"\t\"16 APR 2020\"\t\"20 APR 2020\",\"Dates List\",\"SG0020102 \"\t\"17 APR 2020\"\t\"16 APR 2020\"\t\"20 APR 2020\",\"Dates List\",\"SG0020102-COB \"\t\"17 APR 2020\"\t\"16 APR 2020\"\t\"20 APR 2020\",\"Dates List\",\"US0010001 \"\t\"17 APR 2020\"\t\"16 APR 2020\"\t\"20 APR 2020\",\"Dates List\",\"US0010001-COB \"\t\"17 APR 2020\"\t\"16 APR 2020\"\t\"20 APR 2020\""}

The excel tool uses this mechanism to execute the actions. It will execute an HTTP-POST request, and read the response. The VBA code for this is:



To read the response, it will load the ResponseText property of the request, and parse it to find the OFS response. You can find the code in the Excel tool when editing macro's, in the routine LaunchOFSCommandTAFJ in module1.

### TAFJ Application User

As you can see in the above example, when opening the HTTP connection, you need to pass a username and password. These are not the same as the T24 user/password that you have to provide inside the OFS command, and which is a user with which you can login to BrowserWeb, typically INPUTT/123456.

To access the Rest Services for TAFJ, you need a TAFJ application user, and that user needs to be created in the environment that you are targeting. For local modelbanks, there are scripts for managing the users in the jBoss installation (typically “add-user” script in jboss/bin). Please see the appendix, section 7.1, for a detailed explanation on how to create such a user, in short the process should be to get it working on a UTP installation in Windows:

1-     Go to the directory /jboss/bin

2-     Run the command “add-user.bat”

3-     Select “b”

4-     Type username

5-     Type password

6-     In groups type “t24user,TAFJAdmin” in exact case

7-     When asked as a slave type no

There should be no need to restart jBoss after adding the user. If, after adding the user, you get the following response:

<html><head><title>Error</title></head><body>Unauthorized</body></html>

Then the user was not created correctly, or the T24 installation is not correct (typically, using an installer version, not a UTP-version).

The username and password need to be provided in the “installation” worksheet (see chapter 4)

# Common features

Only three worksheets communicate with T24 for now, the parameters worksheet (where you can install the enquiries needed for the tool to work), the request worksheet (that extracts translatable elements), and the translate worksheet (that installs the translations). On these sheets, and any other that might be added later, some parameters will be needed to know where to connect to.

## General configuration

On each of these worksheets, you will find the following fields that are required to connect:



### environment

In the environment, you need to specify the address of the environment where you connect to if you were to login with the browser, and you need to put in the address the part of that address up to “BrowserWeb”. In essence, you will provide the servername or IP-address, and the port.

### T24 User

You will also need to provide the user name, password and company of a valid T24 USER record that has the necessary privileges to extract the data from technical appllications (like VERSION and ENQUIRY and EB.DICTIONARY), and write back the translations to them. A regular bank user profile is normally not enough, you would need an IT-admin profile for this. In the tool, you will need to specify the LOGIN name, not the USER recordID. Typically INPUTT/123456/GB0010001

### Max

This tool permits you to extract or translate items in bulk. To avoid very long-running commands, the “max” parameter limits the number of OFS commands in one request. It is recommended to set this to 20 to start with, and increase with familiarity if needed.

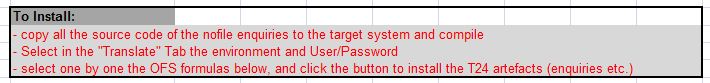
### Function

This can take two values, VALIDATE and PROCESS, and is copied as is into the OFS command. VALIDATE will not save changes to T24 records, but still retrieve the result of a simulated command. PROCESS will change the data in T24.

# Installation

To use the translation tool, some enquiries (mostly nofile enquiries) need to be installed in the Target environment used for extraction (which can be a development or test server). These enquiries are all used for the “request” worksheet, not for the “translate” functionality.

For a nofile enquiry, you need a compiled routine, a STANDARD.SELECTION record and the ENQUIRY record. You will find all these in the parameters worksheet. To install the enquiries, follow the instructions on the worksheet:



## Transferring and compiling source code

As of now, there are 5 routines to install and compile:

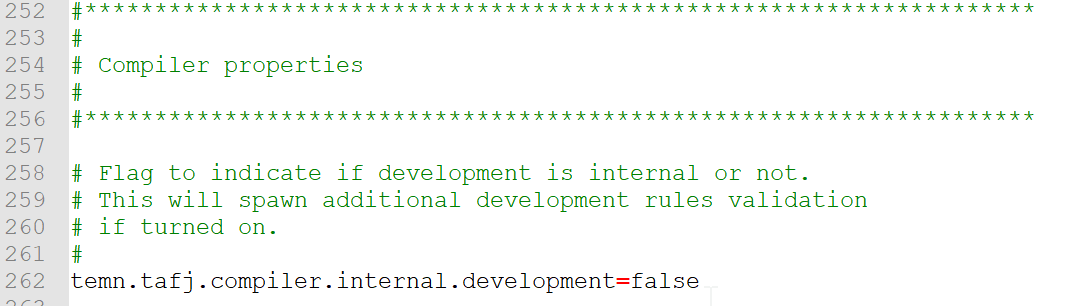
* E.NOFILE.BLM.TRANSLATION.EXTRACT.b
* E.NOFILE.TRANSLATE.APPL.b
* BLM.TRANSLATION.RETRIVE.b
* E.MENU.EXPORT.b
* BLM.EXTRACT.EB.DICT.b

The code is in the parameters worksheet as an embedded file, please copy them to a local path and give the file the correct name (the name of the routine with .b extension). You can either FTP or SFTP them to the target environment or copy them if it is a local or windows installation, to any xxxx.BP, for example TRANSLATE.BP directory. Start a jShell (with putty for example, for a Model Bank UTP installation there should be a command.bat script in the Temenos root directory) session, and compile the routines with a commandline like:

tCompile TRANSLATE.BP\E.NOFILE.BLM.TRANSLATION.EXTRACT.b

please indicate the correct path to the file you copied (if you started command.bat, the current directory will be the T24 home directory, where the TRANSLATE.BP directory was created).

If you have an error that specifies that $PACKAGE is not indicated, make sure that in the tafj.properties file (that you will find in TAFJ\conf) the temn.tajf.compiler.internal.development key is set to false. You will not need to restart jBoss, but you need to recreate a shell commandline with command.bat or putty.

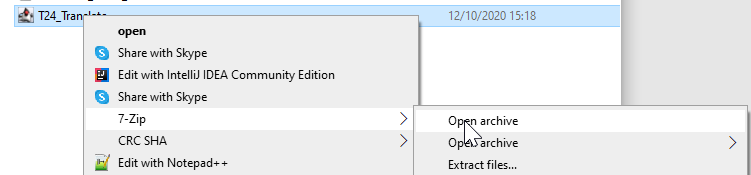
.

*Part of the tajf.properties configuration – the compiler internal.development key is set to false*

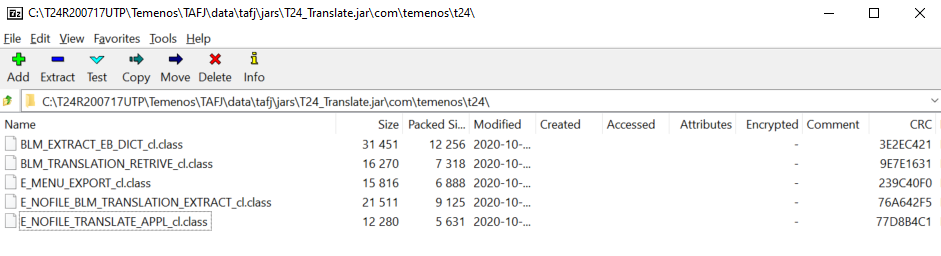
## Deploying the routines to jBoss

### Find the compiled routines

Upon compilation, for each of your routines, a class will be created and packaged in a Java .jar file. Typically, you will find the classes in the .\Temenos\TAFJ\data\tafj\classes\com\temenos\t24 path, and the jars in the . \Temenos\TAFJ\data\tafj\jars path, in the T24\_NoPackage.jar. You can view its contents using 7zip:



Which will show you the classes contained in the jar (go to com/Temenos/t24):



### Rename the jar, and copy it to the application server

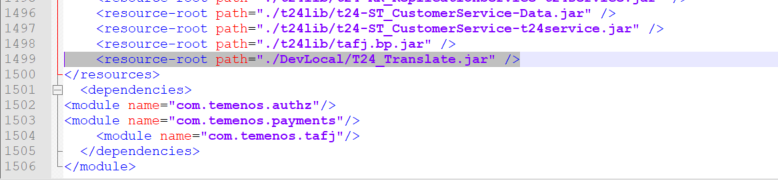
Rename the jar to something identifiable (for example: T24\_Translate.jar)

in a UTP jboss installation, the Temenos core jars would be in the following path: .\Temenos\jboss\modules\com\temenos\t24\main\t24lib

To avoid confusion, create a separate folder for your own developments (like .\Temenos\jboss\modules\com\temenos\t24\main\LocalDev) and copy your renamed jar to this folder.

### Add the new jar to the module list

In the “main” folder (just above t24lib, .\Temenos\jboss\modules\com\temenos\t24\main), there is a file module.xml, where the new jar should be added to the end of the “resources” list:



### Subsequent updates

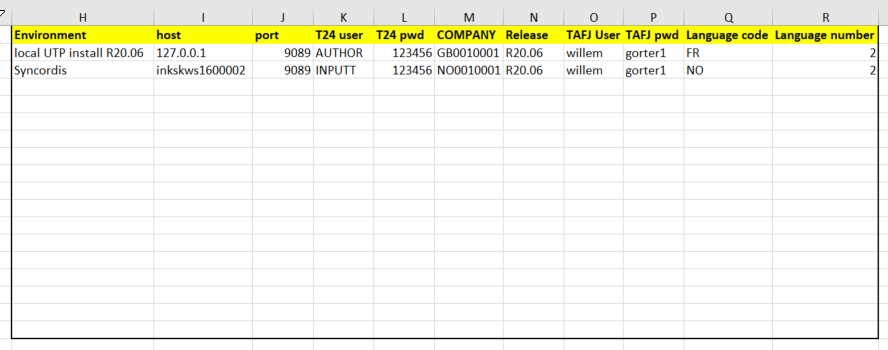
You can delete classes from the jar and replace them with the classes from \Temenos\TAFJ\data\tafj\classes\com\temenos\t24, but jboss should be stopped before the copying action, and then restarted to take the new classes into account

## Setting T24 environment variables

### Creating a list of environments

You can use the Parameters tab to describe the needed variables for multiple T24 environments, and at run time pick the environment nickname and switch between environments.

The T24 Data that need to be entered are:

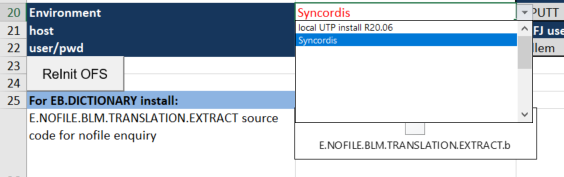


Where the hostname can be found in the browser link just after the http://

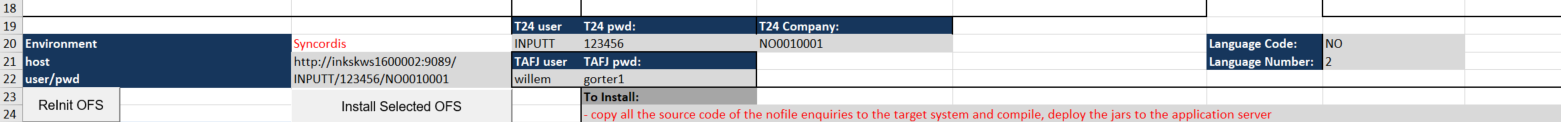
For the TAFJ User and password, please see section 2.2

### Choosing the environment for execution

To choose the environment for execution, a nickname needs to be chosen in cell B20:

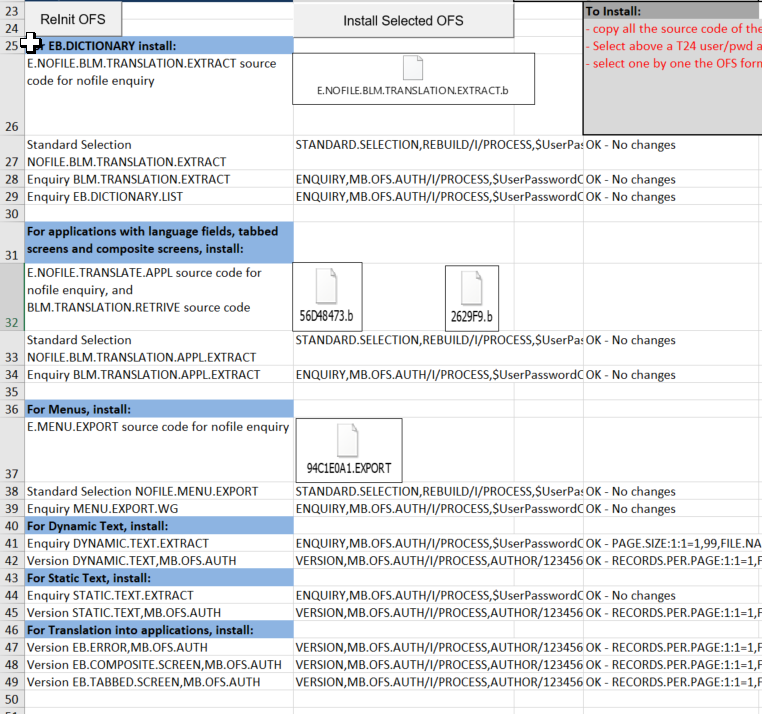


This will load all the environment data from the list.



## Installing the T24 records for STANDARD.SELECTION and ENQUIRY

You can use the Parameters tab to easily install the T24 data elements:



Just select in column B (below the INSTALL button) the cells with the elements that you want to install, and click on the INSTALL SELECTED OFS button. The Excel will post the OFS that is in column B to the environment that you defined just above (don’t forget to adapt it!), and it will give you the result in column D just next to it. If you “Install” again, or if the elements are already installed, the OFS Result would normally be “ERROR, LIVE FILE NOT CHANGED”, but this will be replaced with “OK – No changes” in the result. In any other case, it will give the resulting OFS return:



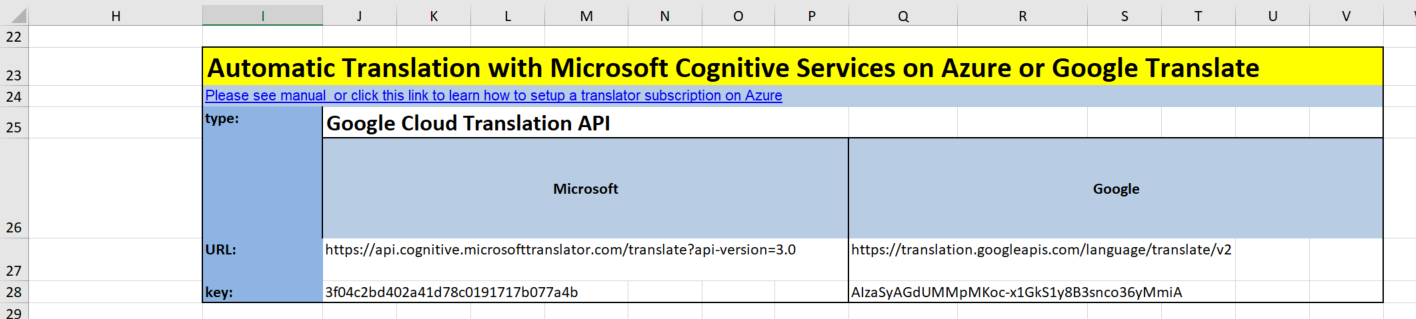
Please start with one Cell and expect the first installation to fail (due to spelling mistake in the environment or the user/password, or unavailability of the TAFJServices/OFSService). Check the result to solve the problems. You can try to execute a Random OFS using the below excel to test connectivity:



The code here is reduced to 20 lines, so it should be easier to find connectivity issues.

## Setting up Automatic translation

To automatically translate English texts to the target language, the tool can call the Microsoft Translator Service hosted in the Azure cloud (Microsoft Cognitive Services) or Google Translate hosted in the Google Cloud. To use this, the excel tool needs a subscription key, which needs to be copied to the Parameters tab, cell J28 (for Microsoft) or Q28 (for Google). Set the translation service you want to use in J25:



To automatically translate English texts to the target language, the tool can call the Microsoft Translator Service hosted in the Azure cloud (Microsoft Cognitive Services). To use this, the excel tool needs a subscription key, which needs to be copied to the Parameters tab, cell J26:

The URL should be the same for everyone, but might evolve in future.

To obtain a subscription key, you need to either create an Azure account or a Google Cloud Platform account:

### Microsoft Azure and Cognitive Services

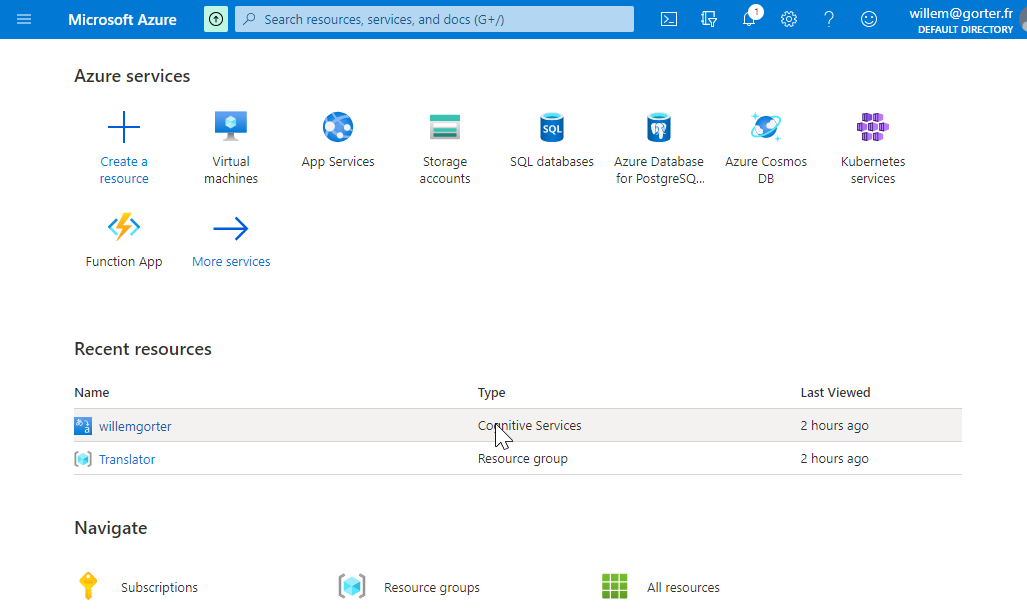
To get a Microsoft Azure Cognitive Services subscription key, you need to create a Microsoft Azure account, and create a Translator resource in the account. To do that, you can follow the instructions described here:

<https://docs.microsoft.com/en-us/azure/cognitive-services/translator/translator-text-how-to-signup>

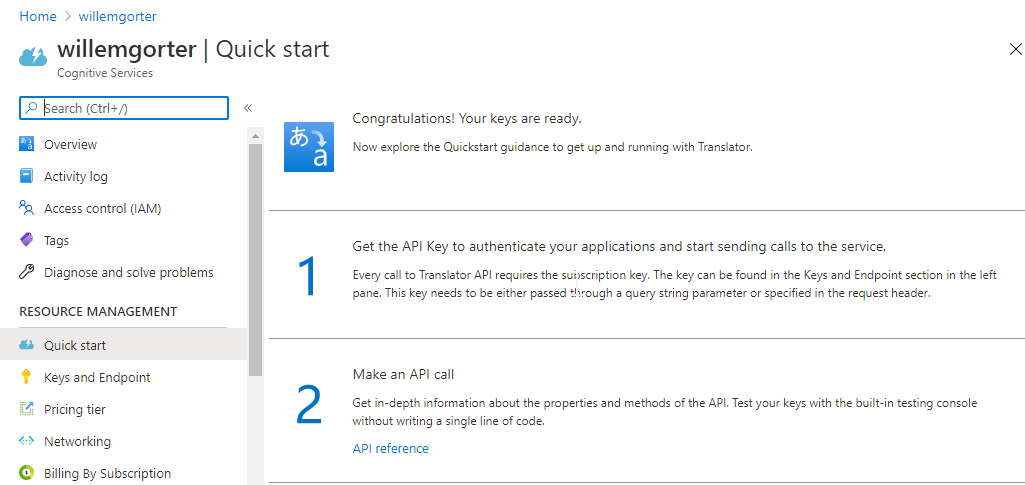
It takes about 5 minutes to sign up and activate the Translator service and obtain your keys, if everything goes well. You need a Microsoft Live account (or create one), as well as a phone (for dual factor authentication), and a credit card (or debit card), although nothing is debited.

The “free” pricing plan allows you to translate up to 2 million characters per month, and will not allow you to translate more (and debit your card) until you upgrade to “pay as you go”.

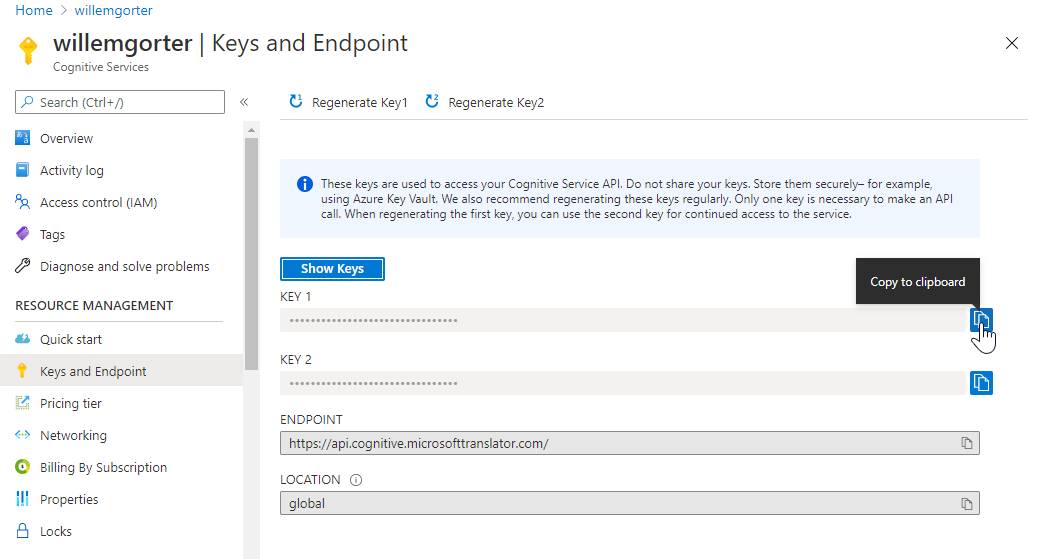
After activation of the Translator service, the Azure home screen looks like this:



Clicking on your resource (willemgorter for me), brings you to the resource menu:



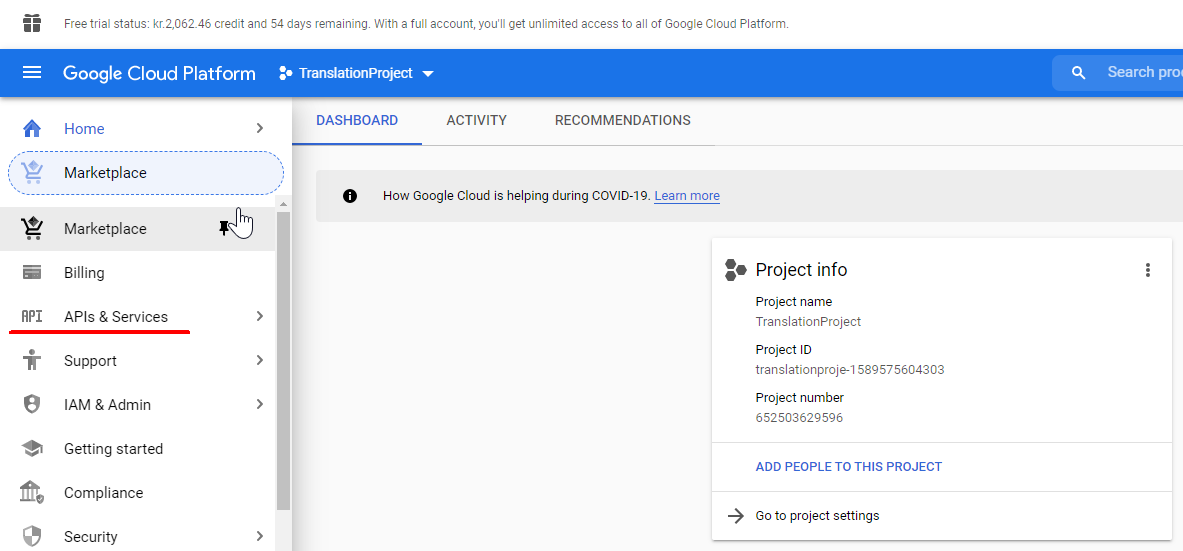
And when clicking on “Keys and Endpoint” in the left-hand menu, you can access and copy the key for the excel tool:

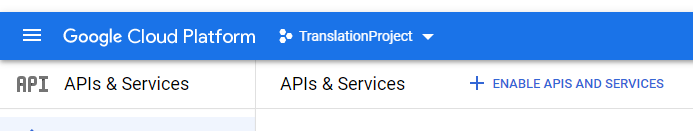


After copying the key to the Parameters tab, you can translate English texts retrieved from T24 Transact, see section 5.5.1

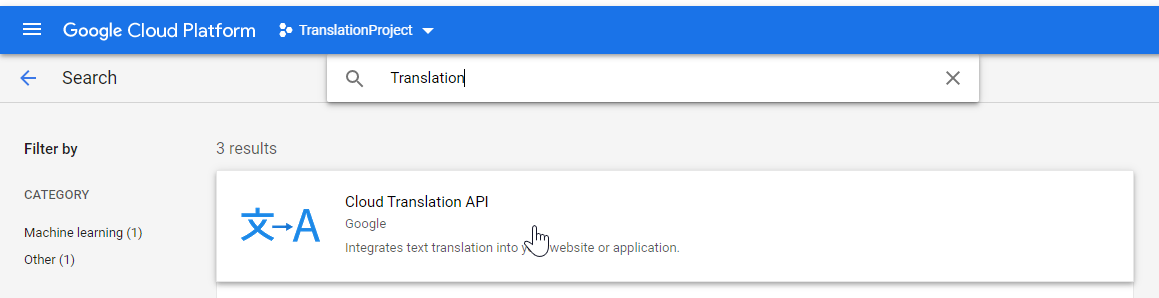
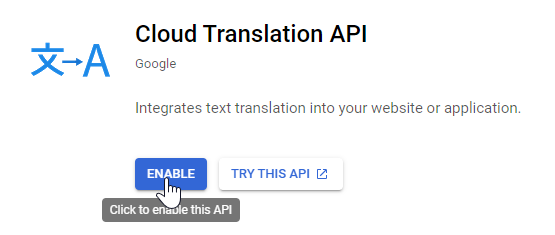
### Google Cloud Platform and Cloud Translation API

To enable translation with Google, you need to create a Google Cloud Platform account. You need a google account to login to Google Cloud Platform. From the Platform Console screen, you need to create a project. After that, choose “API and Services” from the main menu, and click “+ enable APIs and Services”:

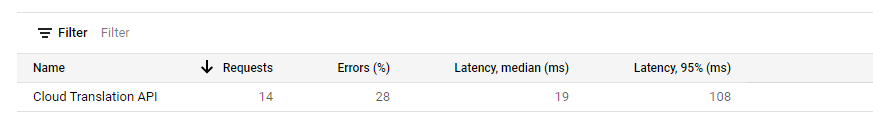




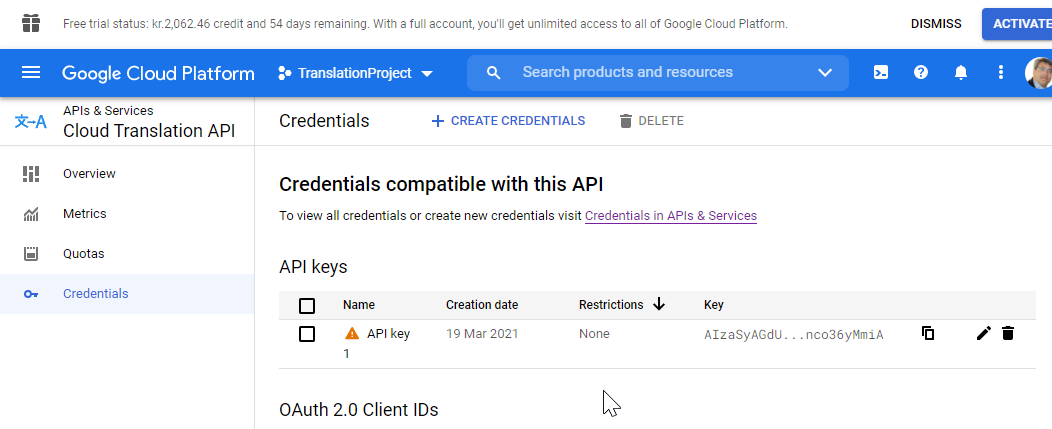
Search for “Translation” and choose to enable Cloud Translation API:

On your dashboard, you should have the API enabled:



Click on the service, and choose “credentials” from the main menu. The API key will be displayed and you can copy it to the clipboard using the icon, and then paste it into the excel.:



# Usage

After installation of the enquiries, the translation process should be as follows:

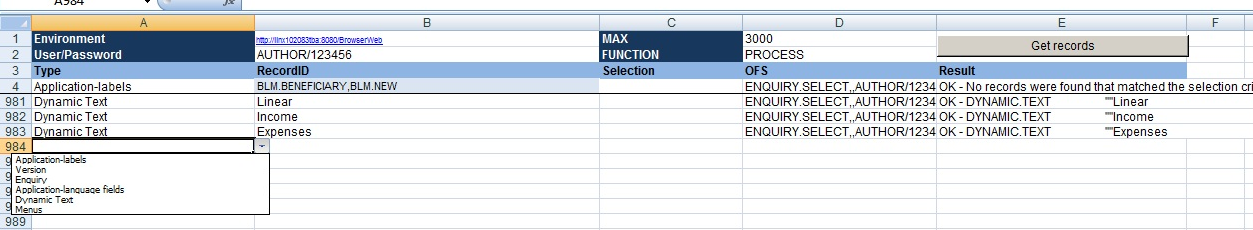
1. Identify T24 artefact to be translated (VERSION, ENQUIRY, …)
2. Extract the translatable elements
3. Copy the translatable elements to the Translation sheet
4. If necessary, correct English text
5. Upload English corrections
6. Translate to your configured language
7. Upload translations
8. TEST if the artefact still works

Especially in enquiries, there might be codewords in translatable fields. Normally, these would be in capitals, and that should be a sign that it shouldn’t be translated maybe.

## Extraction

In the tab "Request", you can copy the list of the elements you want to extract. Please make sure the first two lines are correctly initialised (environment, user/password)

The MAX parameter puts a limit on the number of retrieved records. Especially when extracting from Application-language fields, this can eliminate some aggravation if you make an error in your filter or forget it. The FUNCTION parameter here does not do anything, as you are not making any changes in this worksheet, just retrieving data.



### Types of requests

You can specify different types of requests:

* Application-labels
* Version
* Enquiry
* Application-language fields
* Dynamic Text
* Menus
* Tabbed screens
* Composite screens
* Static Text

Application-labels: this extracts for an application all the fields, so you can create a default translation for all the versions for that application in EB.DICTIONARY. After that, you do not need to translate any versions for that application, except in the specific context if the translation should be different (for example if you are using a field for a different purpose than originally intended)

For an application, there is no default label in English, so we need to provide also the English translation for EB.DICTIONARY. By default, the tool puts in the English label the technical field name with the dots replaced by spaces.

If you looked sharply, in the above image, the first element should not have as RecordID “BLM.BENEFICIARY,BLM.NEW”, which is a VERSION name, but just BLM.BENEFICIARY. The tool will ignore the comma and anything after it.

For a local table, it is best always to start with the application, because you will not have to translate any versions that you create on it after that.

Version: It will, like for Application-labels, extract for a specific version all fields that are present on the version, including any translation that is present. From there, we can insert it in EB.DICTIONARY

Enquiry: It will extract all translatable fields for the enquiry. It will take “operation” fields if between quotes, but this is dangerous, as they may be used for keywords, variables or version names in “IF” decision structures within the enquiry. Requires testing!

Application-language fields: this extracts from a specific application all records in the selection, and for all those records all fields that are of type Language. This is not for field labels on versions, but for example a description field in AA.ACTIVITY, or a virtual table in EB.LOOKUP. if nothing is specified in Selection column, it will extract all records, which can be a lot for example for EB.LOOKUP. Adjust the MAX elements when working with this type.

Dynamic Text: this extracts Dynamic Text records for translation. Dynamic Text is the “old” way of translating strings, and can still be used particularly for hardcoded strings in programs, using the TXT(string\_to\_translate) function. You can specify the RecordID of the Dynamic Text record (it has to be live first), and you can include three dots for wildcards.

Alternatively, you can specify a numeric key, and T24 will interpret it as a date. It will take all records for that date: 20200825 would retrieve all records with DATE.TIME starting with 20200825, so all records created on that day.

Menus: When retrieving a menu, the entire branch of submenu’s will be retrieved as well. Alternatively, if you do not want the submenus, you can use “Application-language fields” with Record “HELPTEXT.MENU” and as filter criteria the name of the submenu record (for example “USER.MENU”).

Tabbed screens: In tabbed screens (EB.TABBED.SCREEN), the only translatable elements are the tab names

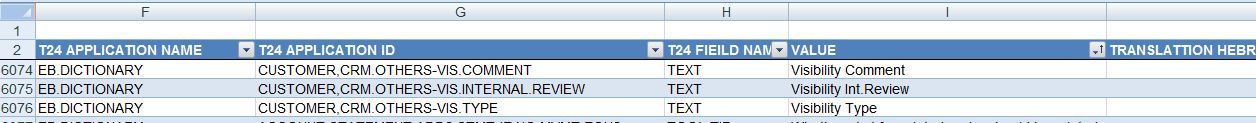
Composite screens: In composite screens, only the title bar is translatable

Static Text: In the STATIC.TEXT application, some elements can be translated like months and days of the week, but please be careful, you can mess up T24 very quickly. In general, you need to translate the record GLOBAL.TEXT at least.

### Launching a request

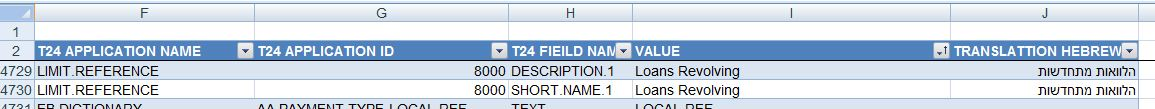
To launch a request, select at least 2 cells, and click the button. The lines included in the selection will be executed, and the results can be found in the Results column – that is, to find out if there was any error.

If the extraction was correct, the usable result can be found in the Extract sheet. The lines will be added to anything that was already in the sheet, it won't overwrite, you can choose either to delete regularly from the Extract sheet or to leave everything. You will need to use columns F to J:

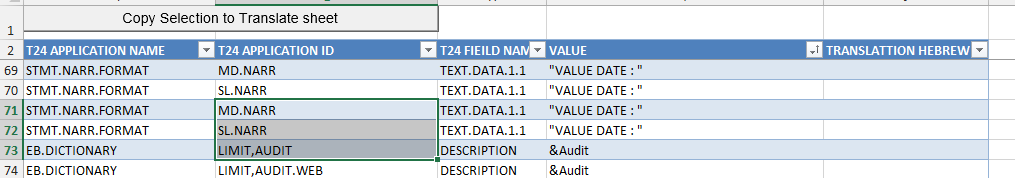


The tool will determine for you where to store the translation (in the above example, for a version, in EB.DICTIONARY), and generate the recordID and the field name (and multivalue position if necessary) where to store the translation.

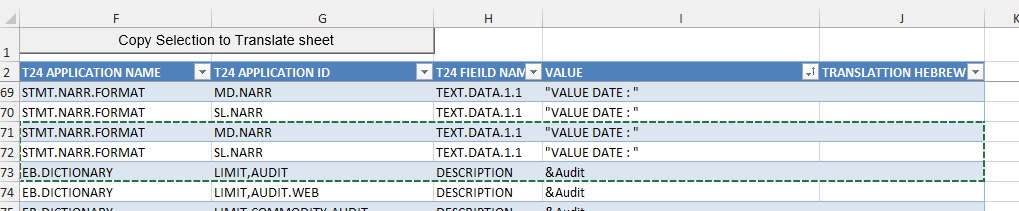
Sometimes, when a translation is already present in the system (when the developer has already put the translations in, or the artefact was already translated before), the tool will also retrieve the translations:



## Copy the translatable elements

If you are happy with what was extracted, you can copy the five columns F to J for all the rows you extracted to the “translate” worksheet, where the real work happens. You can use a regular Excel copy, or select cells within the rows you want to copy 

and use the “copy selection to Translate sheet” button. This will copy the 5 columns for your selection of rows to the first empty cell at the bottom of the translate worksheet.



## Correct English text

In case the English text is not correct (or if the English text contains target language text…), the first thing to do is fix the English text. In the column “value” you can correct the English.

## Upload corrected English text

To upload your correction to T24, select more than one cell, where all the lines in the selection will be “executed”. If you want to correct one single value, select more than one cell on the same line. Make sure that the column “already translated” is empty, it will not execute any line within your selection that has that column filled (to avoid duplicates which return an error “LIVE RECORD NOT CHANGED”)

When you have made your selection, click on the “Correct English” button to upload the changes to T24. The tool will return the result of the update in the Result column. If an error is returned, the OFS command column can also be used to find out what went wrong.

## Translate to Local language

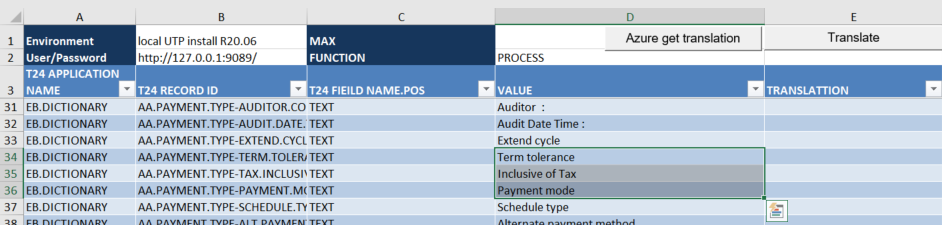
In the Translation column, you can enter the translation of the value column. When the translation is the same as the English text (for example when the value is numeric), you do not need to enter anything, it need not be translated.

Texts that begin with an ampersand (like &Account) can be translated without taking the ampersand into account. This was a way in the DeskTop application to navigate to the field, and is no longer in use. This only goes for Version fields! Ampersands in Error Messages are variables that will be replaced by values on execution, please keep them and put them in the correct place in the translation.

### Automatic Translation with Microsoft Translator or Google Translate

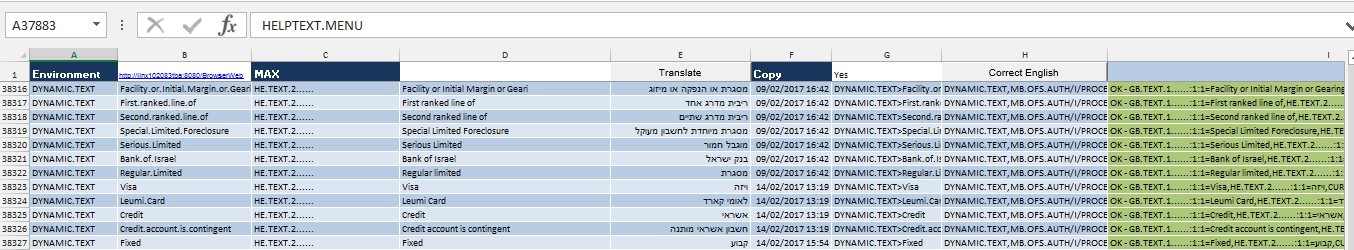
Microsoft has a service that translates text from any language, essentially the same as Google Translate. Both those service can also be called through an API, and this has been implemented in the translation tool. 2M characters per month are free translations for Microsoft, with Google you keep your initial credit. Please see section 4.5 on how to setup the service.

To translate text, from the Translate tab, select at least two cells (either horizontally or vertically), where the selection should embrace all lines that need to be translated. Any line for which the “translation” column is not empty will not be (re-) translated, so delete the text in that column for your selection if you want to use the Automatic Translator service to re-translate. After having made your selection, click the “Azure/Google get translation” button to translate – the language code will be taken from the configured environment (in the Parameter tab)

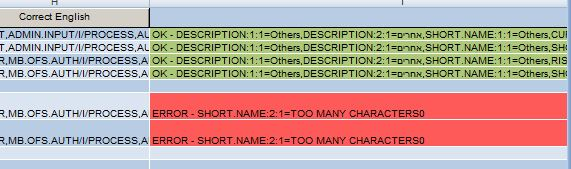


## Upload translations

The same selection method applies as for correcting the English texts. After selecting, click on the Translate button. The columns “Already Translated” and “Packman list” will be filled, the first with the current date and time (so you can select all “new” translations for packaging), the second with the recordID to be used for Packman deployment



The result will show in the Result column, which will show Red if there is an error:



For Versions, some fancy sidestepping is done, compliant with EB.DICTIONARY methodology:

* The tool first checks if a translation exists for the same field in the application. If it is not the case, it will translate the field for the application, so that all other version will profit from the translation
* If the translation for the application exists and is the same as the one to be uploaded, no action is undertaken (the version will already show the correct translation
* If the translation for the application exists and is different from the translation for this application, it will upload the translation for a version-specific translation, other versions will continue to show the translations from the application. To correct the translation of the application, extract the application fields (type Application-labels) and translate the field there

This is very important if you want to correct that translation! When you uploaded the first time, if the translation went to the application, then after correction, the translation will end up in a version-specific record. Your version will show correctly, but all other versions for that application will still contain the error.

Therefore, VERSIONs should only be translated if the translation is context specific, different from what the application intended. Preferably, the application should be translated.

## Test and debug

Please test and debug translated artefacts! Especially enquiries can stop working after translation, as “fieldnames” can be used as containers for IF-constructs, and especially to construct drilldown commandlines. The worst is when CURRENT variables are populated by a value which is translated, because the User will not even now where to look.

If these “fieldnames” are translated, they won’t work anymore. As a general principle, words in enquiries in CAPITAL should not be translated, but it is not entirely 100% watertight, and Enquiries are not the only vulnerable applications (AA.ACTIVITY can be nasty).

# Disclaimer

This Excel tool does not claim to be industrialised for common usage, there might be errors in the tool that are not handled. The Visual Basic for Applications (VBA) that is used for the tool is easy to read, and any T24 developer should be able to debug. Please do not hesitate to sollicite me in case of:

* Unsolvable problems
* Ideas for improvement

Willem Gorter  
[ewgorter@temenos.com](mailto:ewgorter@temenos.com)

Support will only be provided if I am available, this tool does not come with any support guarantees from Temenos. Please do not ask questions without having tried to find the answer in this manual.

## Translation of Helptext

Helptext does not reside in T24, but in XML-files.

There is a separate tool (UserTool) that permits retrieval in excel, translation, and creation of the translated helptext xml files. Please contact Willem Gorter for details

# Appendix

## Creating a TAFJ user on a local T24 installation

Courtesy of Karim Farid.

For release 2020.03 of Transact at least, please download the UTP version, not the installer version, as it only works on the UTP release!

Steps to get it working:

1. Go to the directory /jboss/bin
2. Run the command “add-user.bat”
3. Select “b”
4. Type username
5. Type password
6. In groups type “t24user,TAFJAdmin” in exact case
7. When asked as a slave type no
8. Try again with the new user.

