SDL Trados Studio Automation Kit (STraSAK)

Preface

Quite a number of projects use more-or-less fixed process and structure:

- TMs are stored in a fixed location
- handoffs use fixed folders structure
    - *folder with prepared source files has always the same name...*
    - *folder to create the Trados Studio project too...*
    - *folder to store packages for- and from translation too...*
    - *etc...*
- Trados projects are named using a fixed scheme *(e.g. same as handoff folder)*
- source and target languages are (almost) always the same
- and so on...

So... it should not be necessary to enter manually all these parameters required for Trados Studio operations over and over again... something should be able to use all this known data and do everything automatically...

And that's the purpose of this automation kit – to allow engineers to automate common Trados-related operations using simple scriptable commands:

Command	Description
New-Project	Create new project – optionally based on project template or another project – in specified location, using specified source and target languages and TMs from specified location. Get source files from specified location and automatically convert them to translatable format and copy them to target languages. Optionally also apply PerfectMatch, pre-translate and analyze the files, saving results to Trados 2007-formatted or Excel log.
PseudoTranslate	Pseudo-translate specified project, using specified pseudo-translation options. Optionally also export the pseudo-translated target files.
Export-Package	Create translation packages from specified project, using specified package options, and save them to specified location
Import-Package	Import return packages from specified location in a specified project
ConvertTo-TradosLog	Convert Studio XML-based report to Trados 2007-formatted log
Export-TargetFiles	Export target files from specified project to specified location
Update-MainTMs	Update main TMs of specified project with content from the project target language files
New-FileBasedTM	Create new translation memory in specified location, using specified options
Export-TMX	Export one or more Trados Studio translation memories to TMX, optionally applying a filter
Import-TMX	Import content from TMX file in a specified TM, optionally applying a filter

Contents

Technical info

Automation is based on SDL PowerShell Toolkit (https://github.com/sdl/Sdl-studio-powershell-toolkit) with own extensive enhancements and customizations.

The kit consists of two parts:

– **PowerShell modules** containing the actual automation and supporting functions

– **Windows batch wrapper script** for easy integration of the PowerShell functions to batch scripts

Installation and setup

0) **Pre-requisite**: Windows PowerShell 4.0 or newer installed
   This may be required only for Windows 7 and 8.
   Windows 8.1 has PowerShell 4.0 already built-in, Windows 10 has PowerShell 5.0 already built-in.
   Download PowerShell 4.0: https://www.microsoft.com/en-us/download/details.aspx?id=40855
   Download PowerShell 5.1: https://www.microsoft.com/en-us/download/details.aspx?id=54616
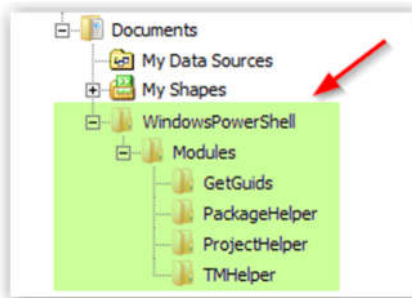
1) **Create "WindowsPowerShell" folder** in your Documents folder
   (i.e. the result will be like "C:\Users\<YourProfile>\Documents\WindowsPowerShell")
   NOTE: If your Documents folder is moved to different location, create the "WindowsPowerShell" folder there.

2) **Copy the entire "Modules" folder** (including the folder structure) **to the created "WindowsPowerShell" folder**.
   The resulting structure should look like this

3) **Copy the `TS2015.cmd` (`TS2017.cmd, TS2019.cmd, TS2021.cmd`) wrapper** script(s) to any preferred location **and add the location to your PATH environment variable**, so that you can the script without specifying its full path.
   See https://www.java.com/en/download/help/path.xml for more information on PATH variable and how to edit its content in different operating systems.

   Note: Make sure to <u>add</u> the location to the existing value of PATH (separated by semicolon)... do not <u>replace</u> the existing value, otherwise some applications or your system may stop working properly!

   Optionally you can put the wrapper script to a location which is already listed in the PATH variable (e.g. C:\WINDOWS)... but that may be uncomfortable, depeding on particular system setup, etc.

That's all... the automation kit is now ready for use!

Usage

From Windows batch script

Call the wrapper script with desired action command and its parameters as command line arguments:

```
call TS2017 New-Project -Name "My project" -Location "D:\My project" ...
```

If the wrapper script location is not listed in PATH environment variable, you need to use full path to script:

```
call "C:\My scripts\TS2017.cmd" New-Project -Name "My project" -Location "D:\My project" ...
```

From PowerShell script

Call the desired PowerShell function with corresponding parameters directly from your PowerShell script:

```
New-Project -Name "My project" -Location "D:\Projects\My project" ...
```

From other scripting languages

Use the language's appropriate method to call either the Windows batch wrapper or the PowerShell function.

Real project usage examples

Following batch scripts show examples of usage in real production.

Target languages list for "CreateProject", "ExportPackages" and "ExportFiles" scripts is passed as script parameter, e.g.:

```
03_CreateProject.cmd "de-DE,fr-FR,it-IT"
```

---- ✂ ---- **03_CreateProject.cmd** --------
```
@echo off
set TARGETLANGUAGES=%~1
for %%D in ("%CD%") do set "PROJECTNAME=%%~nxD"
call TS2017 New-Project ^
     -Name "%PROJECTNAME%" ^
     -SourceLocation "02_Prep" ^
     -ProjectLocation "03_Studio" ^
     -LogLocation "04_ForTrans" ^
     -TargetLanguages "%TARGETLANGUAGES%" ^
     -TMLocation "X:\Projects\Contoso\_TMs" ^
     -ProjectTemplate "X:\Projects\Contoso\_Template\Kitbag.sdltpl" ^
     -Pretranslate -Analyze
```

---- ✂ ---- **04_ExportPackages.cmd** --------
```
@echo off
set TARGETLANGUAGES=%~1
call TS2017 Export-Package ^
     -ProjectLocation "03_Studio" ^
     -PackageLocation "04_ForTrans" ^
     -TargetLanguages "%TARGETLANGUAGES%" ^
     -IncludeMainTMs ^
     -IncludeTermbases
```

---- ✂ ---- **05_ImportPackages.cmd** --------
```
@echo off
call TS2017 Import-Package ^
     -ProjectLocation "03_Studio" ^
     -PackageLocation "05_FromTrans"
```

---- ✂ ---- **06_ExportFiles.cmd** --------
```
@echo off
set TARGETLANGUAGES=%~1
call TS2017 Export-TargetFiles ^
     -ProjectLocation "03_Studio" ^
     -ExportLocation "06_Post" ^
     -TargetLanguages "%TARGETLANGUAGES%"
```

Known Issues

"log4net:ERROR: XmlConfigurator..." message displayed each time automation is started
This is Trados Studio API bug. It's just a cosmetic issue and does not influence automation functionality.

Out Of Memory error during analysis or package creation
This seems to be caused by some weird memory leak in Studio API if a huge TM is used for analysis or for creating Project TM (either during project creation, or when a "Create new TM" package creation option is used), which may cause Out Of Memory exception.
As a workaround you can try the `-PerLanguage` parameter. Or you can try to use smaller/less TMs. Or both.

Automation fails if folder- or file name contains square brackets
This is PowerShell bug and there is currently no reasonable workaround... except for removing there characters from folder/file names.
You can vote for fixing this bug at following links:
https://windowsserver.uservoice.com/forums/301869-powershell/suggestions/15976273-powershell-fails-to-start-if-path-contains-bracket
https://windowsserver.uservoice.com/forums/301869-powershell/suggestions/11088702-start-process-doesn-t-work-if-in-a-directory-name
TIP:
Avoid using weird characters like ampersand, semicolon, percent sign, tilde, round-/square-/curly brackets, etc. in folder and file names to prevent various weird issues.

Commands

New-Project

Creates new Trados Studio project with specified name in specified location, using specified source and target languages.
Project can be optionally based on a project template or other reference project.
Project location is created automatically; existing location is emptied before creating the project.
Translation memories (*.sdltm) are searched in specified location according to source and target languages.
Source files are added from specified location recursively including folders.

Scan, *Convert to Translatable Format* and *Copy to Target Languages* automatic tasks are run during project creation.

Optionally also *PerfectMatch* and/or *Pretranslate* and/or *Analyze* tasks can be run.
These can be also run for each target language separately (as opposed to default Studio behavior which runs the tasks for all languages at once, sometimes leading to Studio crashes, out-of-memory exceptions, etc.)

SYNTAX

New-Project -Name <String> -ProjectLocation <Path> -SourceLocation <Path> [-SourceLanguage <String>] [-TargetLanguages <String>] [-TMLocation <Path>] [-LogLocation <Path>] [-ProjectTemplate|-ProjectReference <Path>] [-OverrideTM] [-PerfectMatch <Path>] [-Pretranslate] [-Analyze] [-PerLanguage] [-ExcelLog]

PARAMETERS

Parameters marked red are mandatory.
Parameters marked blue are not mandatory if corresponding values are defined in project template or reference.
Remaining parameters are optional.

-Name (or **-ProjectName** or **-PrjName**)	Project name. Must not contain invalid characters such as \ / : * ? " < > |
-ProjectLocation (or **-Location** or **-PrjLoc**)	Path to directory where the project will be created. Any existing content of the directory will be deleted before creating the project. If the directory does not exist, it will be created.
-SourceLocation (or **-SrcLoc**)	Path to directory containing project source files. Complete directory structure present in that directory will be added as project source.
-SourceLanguage (or **-SrcLng**)	Project source language locale code. For locale codes, see https://msdn.microsoft.com/en-us/goglobal/bb896001.aspx
-TargetLanguages (or **-TrgLng**)	Space-, comma- or semicolon-separated list of locale codes of project target languages. For locale codes, see https://msdn.microsoft.com/en-us/goglobal/bb896001.aspx
-TMLocation (or **-TMLoc**)	Path to directory containing Trados Studio translation memories for project language pairs. Directory will be searched for all TMs with language pairs defined for the project and found TMs will be assigned to the project languages. Additional TMs defined in project template or reference project will be retained, unless "OverrideTM" parameter is specified. Note: directory is NOT searched recursively!
-LogLocation (or **-LogLoc**)	Path to directory where Trados 2007-formatted textual logs will be created. Any existing logs will be overwritten.
-ProjectTemplate (or **-PrjTpl**)	Path to project template (*.sdltpl) on which the created project will be based. If this parameter is omitted (and ProjectReference parameter too), default project template set in Trados Studio will be used.
-ProjectReference (or **-PrjRef**)	Path to project file (*.sdlproj) on which the created project will be based.
-OverrideTM (or **-OvrTM**)	Ignore TMs defined in project template or reference project and use only TMs specified by "TMLocation" parameter.
-PerfectMatch (or **-PM**)	Path to directory containing bilingual files to be used for applying PerfectMatch. Directory must contain subdirectories for individual languages and bilingual files in these language subdirectories must be structured exactly the same way as files in project.

-Pretranslate (or **-Translate** or **-PreTra**)	Run pre-translation task for target languages during project creation.
-Analyze (or **-Analyse**)	Run analysis task for target languages during project creation.
-PerLanguage (or **-PerLng** or **-PerLang**)	Run batch task(s) for each target language separately, as opposed to default Studio behavior which runs batch tasks for all target languages at once.
-ExcelLog (or **-Excel**)	Save analysis reports in Excel XLSX format. Reports will be saved to the same location as the Trados 2007 logs.

EXAMPLES

```
New-Project -Name "Project" -ProjectLocation "D:\Project" -SourceLocation "D:\Sources"
```

Creates project named "*Project*" based on default Trados Studio project template in "*D:\Project*" folder;
source files are taken from "*D:\Sources*" folder;
source language, target languages and translation memories are taken from the default project template;
only default Scan, Convert and Copy to target languages tasks are run.

```
New-Project -Name "Project" -PrjLoc "D:\Project" -SrcLoc "D:\Sources" -SrcLng "en-US" -TrgLng "fi
-FI" -TMLoc "D:\TMs" -PreTra -Analyze
```

Creates project named "*Project*" based on default Trados Studio project template in "*D:\Project*" folder, with American English as source language and Finnish as target language; source files are taken from "*D:\Sources*" folder and translation memories are taken from "*D:\TMs*" folder; and runs Pretranslate and Analyze as additional tasks after scanning, converting and copying to target languages.

```
New-Project -Name "Sample Project" -PrjLoc "D:\Projects\Trados Studio Automation\Sample" -SrcLoc
"D:\Projects\Trados Studio Automation\Source files" -SrcLng "en-GB" -TrgLng "de-DE,ja-JP" -TMLoc
"D:\Projects\TMs\Samples" -PrjTpl "D:\ProjectTemplates\SampleTemplate.sdltpl" -Analyze
```

Creates project named "*Sample Project*" based on "*D:\ProjectTemplates\SampleTemplate.sdltpl*" project template in "*D:\Projects\Trados Studio Automation\Sample*" folder, with British English as source language and German and Japanese as target languages; source files are taken from "*D:\Projects\Trados Studio Automation\Source files*" folder and translation memories are taken from "*D:\Projects\TMs\Samples*" folder; and runs Analyze as additional task after scanning, converting and copying to target languages.

ACTUAL RUN EXAMPLE

PseudoTranslate

Pseudo-translates Trados Studio project, allowing to define specific pseudo-translation options.
Optionally also exports the pseudo-translated target files.

SYNTAX

```
PseudoTranslate -ProjectLocation <Path> [-TargetLanguages <String>] [-Deterministic] [-AppendStart <String>] [-AppendEnd <String>] [-AppendToParagraph] [-ExpandBy <Number>] [-Dollar] [-ExportLocation <String>]
```

NOTE: it's also possible to use shorter "Pseudo" alias instead of the full "PseudoTranslate" form.

PARAMETERS

Parameters marked red are mandatory.
Remaining parameters are optional.

-ProjectLocation (or **-Location** or **-PrjLoc**)	Path to directory where the project is located.
-TargetLanguages (or **-TrgLng**)	Space-, comma- or semicolon-separated list of locale codes of project target languages. For locale codes, see https://msdn.microsoft.com/en-us/goglobal/bb896001.aspx If this parameter is omitted, all project target languages are pseudo-translated.
-Deterministic (or **-Determ** or **-Det**)	Generate the same translations every time pseudo-translate is run. Otherwise, random words will be used.
-AppendStart (or **-Start**)	String that will be appended to the start of each pseudo-translation. If this parameter is omitted, default string "_" (underscore) is used.
-AppendEnd (or **-End**)	String that will be appended to the end of each pseudo-translation. If this parameter is omitted, default string "_" (underscore) is used.
-AppendToParagraph (or **-Paragraph** or **-Par**)	AppendStart and AppendEnd strings are applied at paragraph level. Otherwise they are applied at segment level.
-ExpandBy (or **-Expand** or **-Exp**)	Numeric factor by which the length of pseudo-translated content will differ to the source content. If this parameter is omitted, default factor of 1.3 (i.e. 130%) is applied. **NOTE: Number must use decimal point**, not comma!
-Dollar	Use dollar signs to represent characters in the pseudo-translations.
-ExportLocation (or **-ExpLoc** or **-Export**)	Path to directory where to export pseudo-translated files. If the directory does not exist, it will be created.

EXAMPLES

Export-Package -ProjectLocation "D:\Project"

Pseudo-translates all target languages defined in project located in "D:\Project" folder.

Export-Package -PrjLoc "D:\Project" -TrgLng "fi-FI,sv-SE" -ExpLoc "D:\Export"

Pseudo-translates Finnish and Swedish languages from project located in "D:\Project" folder and exports the pseudo-translated files to "D:\Export" folder.

ACTUAL RUN EXAMPLE

```
Pseudo-translating...
  100%  Completed
Task Pseudo-translate successfully completed.
Done

Exporting files...
  100%  Completed
Task Export files successfully completed.

Saving project...
Done
```

Export-Package

Exports Trados Studio project packages from project to a specified location, allowing to define specific export options. Packages location is created automatically. Separate package is created for every target language.

SYNTAX

Export-Package -ProjectLocation <Path> -PackageLocation <Path> [-TargetLanguages <String>] [-Task <String>] [-ProjectTM <String>] [-PackageComment <String>] [-IncludeAutoSuggestDictionaries] [-IncludeMainTMs] [-IncludeTermbases] [-RecomputeAnalysis] [-IncludeExistingReports] [-KeepAutomatedTranslationProviders] [-RemoveServerBasedTMs]

PARAMETERS

Parameters marked red are mandatory.
Remaining parameters are optional.

-ProjectLocation (or **-Location** or **-PrjLoc**)	Path to directory where the project is located.
-PackageLocation (or **-PkgLoc**)	Path to directory where the package(s) will be created.
-TargetLanguages (or **-TrgLng**)	Space-, comma- or semicolon-separated list of locale codes of project target languages. For locale codes, see https://msdn.microsoft.com/en-us/goglobal/bb896001.aspx If this parameter is omitted, packages for all project target languages are created.
-Task	Name of manual task which will be associated with files in the package. Allowed values: **Translate** *(default)* **Review**
-ProjectTM (or **-ProjectTMs** or **-PrjTM**)	Option for project translation memory to be included in the package. Allowed values: **None** do not include any project TM *(default)* **UseExisting** include existing project TM **CreateNew** create new project TM
-PackageComment (or **-PkgCmt**)	Optional short comment to be included in package
-IncludeAutoSuggestDictionaries (or **-IncludeAutoSuggest** or **-IncAS**)	Include AutoSuggest dictionaries in package
-IncludeMainTMs (or **-IncludeMainTM** or **-IncTM**)	Include main translation memories in package
-IncludeTermbases (or **-IncludeTermbase** or **-IncTB**)	Include termbases in package
-RecomputeAnalysis (or **-RecomputeAnalyse** or **-RecomputeAnalyze** or **-Recompute** or **-RecAna**)	Recompute wordcount and analysis to update cross-file repetition counts and include the recomputed reports in package

-IncludeExistingReports (or **-IncludeExistingReport** or **-IncludeExisting** or **-IncludeReports** or **-IncludeReport** or **-IncRep**)	Include existing wordcount reports in package
-KeepAutomatedTranslationProviders (or **-KeepATProviders** or **-KeepATProvider** or **-KeepAT**)	Keep automated translation providers information in package
-RemoveServerBasedTMs (or **-RemoveServerTMs** or **-RemoveServerTM** or **-RmvSrvTM**)	Remove links to server-based translation memories from package

EXAMPLES

Export-Package -ProjectLocation "D:\Project" -PackageLocation "D:\Packages"

Creates translation packages for all target languages defined in project located in "*D:\Project*" folder;
packages will be created in "*D:\Packages*" folder;
no project TMs, main TMs, termbases, etc. will be included in the packages.

Export-Package -PrjLoc "D:\Project" -PkgLoc "D:\Packages" -TrgLng "fi-FI,sv-SE" -IncTM -IncTB

Creates translation package for Finnish and Swedish target languages from project located in "*D:\Project*" folder;
package will be created in "*D:\Packages*" folder;
project TM will not be included in package, main TM and termbase will be included in package.

ACTUAL RUN EXAMPLE

Import-Package

Imports Trados Studio return package(s) from specified location into a project stored in specified location.

SYNTAX

```
Import-Package -ProjectLocation <Path> -PackageLocation <Path> [-Recurse]
```

PARAMETERS

Parameters marked red are mandatory.

-ProjectLocation (or **-Location** or **-PrjLoc**)	Path to directory where the project is located.
-PackageLocation (or **-PkgLoc**)	Path to either a single package, or a directory where multiple packages are located.
-Recurse (or **-r**)	If **PackageLocation** specifies a directory, all its subdirectories are searched recursively for return packages (*sdlrpx).

EXAMPLES

```
Import-Package -ProjectLocation "D:\Project" -PackageLocation "D:\Packages" -Recurse
```

Imports all translation packages found in "*D:\Packages*" directory (and all its eventual subdirectories) into a project located in "*D:\Project*" folder.

```
Import-Package -ProjectLocation "D:\Project" -PackageLocation "D:\Packages\Batch 1"
```

Imports all translation packages found in "*D:\Packages\Batch 1*" directory into a project located in "*D:\Project*" folder.

```
Import-Package -PrjLoc "D:\Project" -PkgLoc "D:\Packages\Batch 1\Handback_en-US_fi-FI.sdlrpx"
```

Imports single translation package from "*D:\Packages\Batch 1\Handback_en-US_fi-FI.sdlrpx*" folder into a project located in "*D:\Project*" folder.

ACTUAL RUN EXAMPLE

ConvertTo-TradosLog

Converts individual Studio XML-formatted task report(s) to Trados 2007 plain text format.
Output can optionally contain only total summary, omitting details for individual files.
NOTE: Currently <u>only Analyze task report format is supported</u>.

Output files are created in the same location as input files, using input file names and ".log" extension.

When converting single file, the **Destination** parameter can be used to specify complete output path and filename.

SYNTAX
ConvertTo-TradosLog -Path <Path> [-Destination <Path>] [-TotalOnly] [-Recurse]

PARAMETERS

Parameters marked red are mandatory.
Remaining parameters are optional.

-Path	Path to either a single XML-formatted Studio report, or a directory where multiple XML reports are located.
-Destination	Path to a single output text file. Can be used only if Path specifies a single file, not directory. If this parameter is omitted, output file will be created in the same location as input file, using input file name and ".log" extension.
-TotalOnly	Include only total summary (do not include analyses for individual files).
-Recurse (or -r)	If **Path** specifies a directory, convert also all task reports found in subdirectories of the specified path.

EXAMPLES

ConvertTo-TradosLog -Path "D:\Project\Reports\Analyze Files_en-US_fi-FI.xml" -Destination "D:\Finnish analysis.log"

Converts the "*D:\Project\Reports\Analyze Files_en-US_fi-FI.xml*" report to Trados 2007-like plain text log and saves the result to "*D:\Finnish analysis.log*" file.

ConvertTo-TradosLog -Path "D:\Project\Reports" -TotalOnly -Recurse

Converts the all reports found in "*D:\Project\Reports*" directory and its subdirectories Trados 2007-like plain text logs, which will include only total summaries. Converted files will be saved in the directories along the source reports.

ACTUAL RUN EXAMPLE
This command does not produce any output

Export-TargetFiles

Exports target files from Trados Studio file based project to specified location.

SYNTAX
Export-TargetFiles -ProjectLocation <Path> -ExportLocation <Path> [-TargetLanguages <String>]

PARAMETERS

Parameters marked red are mandatory.
Remaining parameters are optional.

-ProjectLocation (or **-Location** or **-PrjLoc**)	Path to directory where the project is located.
-ExportLocation (or **-ExpLoc** or **-Export**)	Path to directory where the target files should be exported.
-TargetLanguages (or **-TrgLng**)	Space-, comma- or semicolon-separated list of locale codes of project target languages. For locale codes, see https://msdn.microsoft.com/en-us/goglobal/bb896001.aspx If this parameter is omitted, target files for all project target languages are created.

EXAMPLES

Export-TargetFiles -ProjectLocation "D:\Project" -ExportLocation "D:\Export"

Exports target files for all target languages defined in project located in "*D:\Project*" folder;
files will be created in "*D:\Export*" folder.

Export-TargetFiles -PrjLoc "D:\Project" -ExpLoc "D:\Export" -TrgLng "fi-FI,sv-SE"

Exports target files for Finnish and Swedish languages from project located in "*D:\Project*" folder;
files will be created in "*D:\Export*" folder.

ACTUAL RUN EXAMPLE

Update-MainTMs

Updates main TMs of Trados Studio file based project with content from the project bilingual target language files.

SYNTAX
Update-MainTMs -ProjectLocation <Path> [-TargetLanguages <String>]

PARAMETERS

Parameters marked red are mandatory.
Remaining parameters are optional.

-ProjectLocation (or **-Location** or **-PrjLoc**)	Path to directory where the project is located.
-TargetLanguages (or **-TrgLng**)	Space-, comma- or semicolon-separated list of locale codes of project target languages. For locale codes, see https://msdn.microsoft.com/en-us/goglobal/bb896001.aspx If this parameter is omitted, main TMs for all project target languages are updated.

EXAMPLES

Update-MainTMs -ProjectLocation "D:\Project"

Updates main TMs for all target languages defined in project located in "*D:\Project*" folder.

Update-MainTMs -PrjLoc "D:\Project" -TrgLng "fi-FI,sv-SE"

Updates main TMs for Finnish and Swedish languages from project located in "*D:\Project*" folder.

ACTUAL RUN EXAMPLE


```
Updating main translation memories...
  100%  Completed
Task Update Main Translation Memories successfully completed.
Done
```

New-FileBasedTM

Creates new file based translation memory using specified source and target language. Optionally also a TM description and further TM behavior options can be specified.
TM can be created
  – using specified name and location
    *(where the name will be used for the internal TM 'display name' and for filename of the created *.sdltm file)*
  – using specified full file path
    *(where the *.sdltm file name will be used for the internal TM 'display name')*

When using only single target language, the specified TM name or .sdltm filename is used as-is.
When using multiple target languages, the source- and target language ISO abbreviations are added as suffix to the specified TM name or .sdltm filename.

NOTE: Trados Studio uses the TM internal display name in user interface (e.g. in translation results pane)!
The automatic setting of TM display name to the file name using the second method may not be always desirable... it may be better to create TM with 'friendly' display name using the first method (and then rename the TM file if needed).

SYNTAX
New-FileBasedTM -Name <String> -TMLocation <Path> -SourceLanguage <String> -TargetLanguages <String> [-Description <String>] [-PassThru] [-FuzzyIndexes <Option>] [-Recognizers <Option>] [-TokenizerFlags <Option>] [-WordCountFlags <Option>]

New-FileBasedTM -Path <Path> -SourceLanguage <String> -TargetLanguages <String> [-Description <String>] [-PassThru] [-FuzzyIndexes <Option>] [-Recognizers <Option>] [-TokenizerFlags <Option>] [-WordCountFlags <Option>]

PARAMETERS

Parameters marked red are mandatory.
Remaining parameters are optional.

-Name (or **-TMName**)	Translation memory name. Must not contain invalid characters such as \ / : * ? " < > |
-TMLocation (or **-Location** or **-TMLoc**)	Path to directory where the translation memory will be created. If the directory does not exist, it will be created.
-Path (or **-TMPath**)	Path of translation memory file (including the ".sdltm" extension!) to be created. **NOTE: Path** parameter cannot be specified together with **Name** and **TMLocation** parameters!
-SourceLanguage (or **-SrcLng**)	Locale code of translation memory source language. For locale codes, see https://msdn.microsoft.com/en-us/goglobal/bb896001.aspx
-TargetLanguages (or **-TargetLanguage** or **-TrgLng**)	Space-, comma- or semicolon-separated list of locale codes of translation memory target languages. For locale codes, see https://msdn.microsoft.com/en-us/goglobal/bb896001.aspx
-Description (or **-TMDesc**)	Optional translation memory description.
-PassThru (or **-PassThrough**)	Return created TM object(s). If single target language is specified, a single TM object is returned. If multiple target languages are specified, an array of TM objects is returned.
-FuzzyIndexes	Set of fuzzy indexes that should be created in translation memory. If this parameter is omitted, default Trados Studio setting for new TM is used.
-Recognizers	Recognizer settings. If this parameter is omitted, default Trados Studio setting for new TM is used.
-TokenizerFlags	Flags affecting tokenizer behavior for translation memory. If this parameter is omitted, default Trados Studio setting for new TM is used.
-WordCountFlags	Flags affecting word count behavior for translation memory. If this parameter is omitted, default Trados Studio setting for new TM is used.

EXAMPLES

```
New-FileBasedTM -Name "Contoso English-German Main" -TMLocation "D:\Projects\TMs" -SourceLanguage
 "en-US" -TargetLanguage "de-DE"
```

Creates "*D:\Projects\TMs\Contoso English-German Main.sdltm*" translation memory with "English (United States)" source language and "German (Germany)" target language and "Contoso English-German Main" internal friendly name.
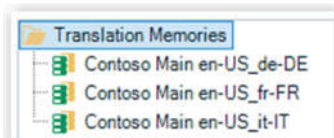


```
New-FileBasedTM -Name "Contoso Main" -TMLocation "D:\Projects\TMs" -SourceLanguage "en-US" -Targe
tLanguages "de-DE fr-FR it-IT"
```

Creates *"Contoso Main en-US_de-DE.sdltm"*, *"Contoso Main en-US_fr-FR.sdltm"* and *"Contoso Main en-US_it-IT.sdltm"* translation memories in *"D:\Projects\TMs"* location, with "English (United States)" source language and "German (Germany)", "French (France)" and "Italian (Italy)" respective target languages. Internal friendly TM names will be set to "Contoso Main en-US_de-DE", "Contoso Main en-US_fr-FR" and "Contoso Main en-US_it-IT" respectively.


```
New-FileBasedTM -Path "D:\Projects\TMs\Contoso_English-German_Main.sdltm" -SrcLng "en-US" -TrgLng
 "de-DE"
```

Creates "*D:\Projects\TMs\Contoso_English-German_Main.sdltm*" translation memory with "English (United States)" source language and "German (Germany)" target language and "Contoso_English-German_Main" internal friendly name.
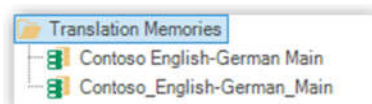
ACTUAL RUN EXAMPLE
This command does not produce any output

Export-TMX

Exports one or more Trados Studio translation memories to TMX file(s).
Optionally, a filter can be applied during export, allowing to export only translation units meeting certain criteria.

Filter is loaded from a filter file with ".sdltm.filters" extension, which can be obtained by exporting it from Trados Studio (from Translation Memories view using "Export filters to file" function).

SYNTAX

Export=TMX [-TMLocation <Path>] [-TMXLocation <Path>] [-FilterFile <Path>] [-FilterName <String>]
 [-Force] [-Recurse]

PARAMETERS

Parameters marked red are mandatory.
Remaining parameters are optional.

-TMLocation (or **-Location** or **-TMLoc**)	Path to either single TM file, or to directory where one or more TMs are located. If this parameter is omitted, current directory is used by default.
-TMXLocation (or **-TMXLoc**)	Path to directory where the exported TMX will be created. If this parameter is omitted, exported TMX will be created in the same location as source TM.
-FilterFile (or **-FltFile** or **-FilterPath** or **-FiltPath**)	Path to TM filter file, containing definition of filter to be used for export. Only translation units matching the filter criteria will be exported. (TM filter file can be created in Studio in Translation Memories view using "Export filters to file" function).
-FilterName (or **-FltName**)	Name of the filter to be used for export. Only translation units matching the filter criteria will be exported. Filter must exist in the filter file specified using FilterFile parameter.
-Force (or **-Overwrite**)	Allows to overwrite any existing TMX file.
-Recurse (or **-r**)	Exports also all TMs found in subdirectories of the specified path.

EXAMPLES

Export-TMX

Exports all Trados Studio translation memories found in current directory to TMX files. Existing TMX files will be preserved.

Export-TMX -TMLoocation "D:\Projects\TMs" -TMXLocation "D:\TMX_Exports" -Recurse -Force

Exports to TMX all Trados Studio TMs present in "*D:\Projects\TMs*" folder and its subfolders.
Exported files will be stored to "*D:\TMX_Exports*" folder. Existing TMX files will be overwritten.

Export-TMX -TMLocation "D:\Projects\TMs\EN-DE.sdltm" -FilterFile "D:\Projects\Filters\Microsoft.sdltm.filters" -FilterName "Word 2016"

Exports a single "D:\Projects\TMs\EN-DE.sdltm" translation memory to TMX.
Exported file will be stored in the same location as source file (i.e. "*D:\Projects\TMs*").
A filter named "Word 2016" from a filter file "*D:\Projects\Filters\Microsoft.sdltm.filters*" will be applied during export.
Exported TMX will then contain only translation units meeting criteria defined in the "Word 2016" filter.

ACTUAL RUN EXAMPLE

Import-TMX

Imports content of TMX file to Trados Studio translation memory.
Optionally, a filter can be applied during import, allowing to import only translation units meeting certain criteria.

Filter is loaded from a filter file with ".sdltm.filters" extension, which can be obtained by exporting it from Trados Studio (from Translation Memories view using "Export filters to file" function).

NOTE: Language pairs of Studio TM and TMX are not checked in any way! User should make sure to not mix languages.

SYNTAX
Import-TMX -Path <Path> -TMXPath <Path> [-FilterFile <Path>] [-FilterName <String>]

PARAMETERS

Parameters marked red are mandatory.
Remaining parameters are optional.

-Path (or **-TMPath**)	Path to translation memory file (including the ".sdltm" extension).
-TMXPath	Path of TMX file (including the ".tmx" extension) to be imported.
-FilterFile (or **-FltFile** or **-FilterPath** or **-FiltPath**)	Path to TM filter file, containing definition of filter to be used for export. Only translation units matching the filter criteria will be exported. (TM filter file can be created in Studio in Translation Memories view using "Export filters to file" function).
-FilterName (or **-FltName**)	Name of the filter to be used for export. Only translation units matching the filter criteria will be exported. Filter must exist in the filter file specified using FilterFile parameter.

EXAMPLES

Import-TMX -Path "D:\Projects\TMs\EN-DE.sdltm" -TMXPath "D:\TMX\English-German.tmx"

Imports "*D:\TMX\English-German.tmx*" file to "*D:\Projects\TMs\EN-DE.sdltm*" translation memory.

Import-TMX -TMPath "EN-DE.sdltm" -TMXPath "English-German.tmx"

Imports "*English-German.tmx*" file to "*EN-DE.sdltm*" translation memory. Both files are located in current folder.

Import-TMX -TMPath "EN-DE.sdltm" -TMXPath "English-German.tmx" -FltFile "D:\Projects\Filters\Microsoft.sdltm.filters" -FltName "Word 2016"

Imports "*English-German.tmx*" file to "*EN-DE.sdltm*" translation memory. Both files are located in current folder.
A filter named "Word 2016" from a filter file "*D:\Projects\Filters\Microsoft.sdltm.filters*" will be applied during import.
Only translation units meeting criteria defined in the "Word 2016" filter will be actually imported into the translation memory.

ACTUAL RUN EXAMPLE


```
ts2017 Import-TMX -Path "DE-DE_███_███████.sdltm" -TMXPath "███████_██ ███_de-DE.tmx"
███████_██ ███_de-DE.tmx -> DE-DE_███_███████.sdltm
TUs processed: 51065, imported: 51065 (added: 8415, merged: 42650), discarded: 0, errors: 0
```

Version history

STraSAK v1.8   **2021-01-13**
+ Added support for Studio 2021

STraSAK v1.7   **2019-10-11**
+ Improved detection of default project template *(it works now correctly with Studio 2019 SR1 and newer, which broke the way it worked reliably for years :( )*
+ **Import-TMX** command now displays separate progress information for each import phase

STraSAK v1.6   **2019-09-15**
+ **Import-TMX** and **Export-TMX** commands can now apply filter during import/export
! fixed "*Multiple ambiguous overloads found*" errors and no progress information displayed during TMX import/export with Studio 2017 SR1 CU13 and newer
+ progress information is now displayed also for Pre-translate Files and Analyze Files automatic tasks
+ **New-FilebasedTM** now optionally returns created TM object(s) using -PassThru switch
+ TM helper functions (Get-TM...) now accept also TM object as input, apart from TM file
∗ *minor internal refactoring*

STraSAK v1.5   **2019-07-03**
+ **New-FileBasedTM** command can now create multiple TMs at once

STraSAK v1.4   **2018-06-29**
+ Added support for Studio 2019

STraSAK v1.3   **2018-02-04**
! **New-Project**: **fixed** v1.2 bug causing **source language code missing in created Trados log file names** *(e.g. `Analyze Files de-DE.log` instead of `Analyze Files en-US_de-DE.log`)*

STraSAK v1.2   **2018-01-29**
! **fixed bug causing Project TM (and analysis report) not being included in created translation package when "Create new project TM" option was used**
+ **Export-Package**: added "Recompute" and "IncludeExisting" aliases for "RecomputeAnalysis" and "IncludeExistingReports" parameters
+ **Export-TargetFiles**: added "Export" alias for "ExportLocation" parameter, to be consistent with pseudotranslation parameters
∗ *some internal refactoring... plus, PowerShell files are now UTF-8 encoded*

STraSAK v1.1   **2018-01-17**
+ added **PseudoTranslate** command

STraSAK v1.0 Final   **2017-12-29**
+ added **New-Project** command features:
     o can also apply PerfectMatch during project creation (with automatic matching of bilingual files)
     o can also save analysis logs to Excel format
     o can run in special "per-language mode" when tasks (pretranslation, analysis, etc.) are run separately for each language – see more info below
+ added/changed **ConvertTo-TradosLog** command features:
     o can now **convert all logs in folder** (incl. subfolders, if needed)
     o can now **specify output file name and location** when converting single file
+ new **Update-MainTMs** command which does exactly what it says ;-)
∗ changed **Import-Package** command feature
     o now does not search input folder recursively by default
     o separate -Recurse switch needed to search for packages also in subfolders
+ new **New-FileBasedTM** command for quick creation of empty TMs
+ new **Import-TMX** command for quick TMX import to TM
+ new **Export-TMX** command for quick TMX export
     o can export all TMs in folder, incl. subfolders if needed
∗ improved progress messages

STraSAK v1.0 beta hotfix 1   2016-12-02

* internal fuzzies now correctly included in Trados 2007 textual logs

STraSAK v1.0 beta   2016-11-21

* Studio project creation, including pre-translation, analysis and textual logs export
+ translation packages creation
+ return packages import
+ target files export

STraSAK Technical Preview   2016-10-10

Basic project creation functionality