

Automating your translation project

How to use project templates & workflow automation in memoQ server



Table of contents

Introduction	3
Getting started	5
Creating a project template	8
Naming patterns	12
Example: Automating file exports	16
Configuring the automated actions	21
Creating a project from the template	27

Introduction

The idea behind project templates and workflow automation in memoQ is a very simple desire to make the project manager more productive. When creating a new project, there is an awful lot of repetition. We wanted to reduce the number of clicks needed before launching a project. We came up with the arbitrary number of 80%. We hoped that a feature that reduces the amount of work on a given task by four-fifths would very likely be welcome.

Since we introduced project templates, we heard good reports from customers who were saving a considerable amount of time, and benefiting from this innovation. However, it is also apparent that many memoQ users are unaware of project templates. With this book, we hope to inform more people of the benefits they can achieve with this technology.

This eBook is aimed at project managers who work with memoQ server. It is an introduction that will allow you to get started with project templates. However, there is some very powerful functionality here. If you take the time to learn working with project templates, you could be rewarded with spectacular process automation.



The chapters in this book cover the following topics:

- WHY PROJECT TEMPLATES This section takes you through the process memoQ went through in coming up with project templates.
- **GETTING STARTED** This section shows you the first steps in designing a project template taking an existing workflow and automating.
- **CREATING A PROJECT TEMPLATE** As the name suggests, this is an introduction to creating a project template.
- NAMING PATTERNS These are crucial to better management of project resources. When you use naming patterns to set names for TMs (translation memories) and other resources, you can keep track of and reuse resources more easily.
- **AUTOMATING FILE EXPORTS** This is just one of the actions you can automate: It allows you to automatically export the file after a trigger has occurred.

Much of the content of this book was originally published as blog posts written by our former colleague, Monika Antunovics. We would like to thank her for her continued support and encouragement.





Getting started

Getting started

You should start by writing out or drawing a diagram of the process you want to automate. It is possible to automate most of the steps in the process. You should decide on what you will and will not automate.

The diagram below is a very simple workflow we have created around a translation project.



You want to do several things with this project. You want to automate as much as you can. You also want to be able to use the same TM, LiveDocs and terminology resources every time you receive a similar job.



You review the workflow, and confirm that the steps in green are the ones you want to automate, those in orange are to be semi-automated, and those in blue are manual steps. You create the project as described <u>below</u>, and automate the steps as described in the section on Automating file exports.



You also want to ensure that when similar projects occur at a later stage, you can find them and leverage the resources involved in previous projects. You do that by using <u>naming patterns</u> and placeholders in the TM names in the template – as in "{Client}-{Subject}-{SrcLangIso2}-{TmTrgLangIso2}-Working". This means that when you create another project with the same client, subject and language pair, it will use the same TM. If you are from a translation company, we suggest that you define workflows for your regular customer, and then convert these to project templates.







Creating a project template

Creating a project template

Start by opening the Resource Console, and clicking the resource type "Project templates":



You can create a new one and then edit it, just like you do with any other resource. When you start editing a project template, it will be a local or online template, depending on whether it's stored on your computer or a memoQ server. However, it is very easy to switch between local and online project settings: just click the eye-shaped icon at the top right corner of the template editor window:





If you click it when editing a local template, you get a new category, "Online project settings", and some other settings in the already seen categories – in particular, the automation options – are expanded:

Another handy feature of this dialog is the "Preview full list of template settings" link at the bottom of the window:







Clicking it will open an HTML page which is suspiciously similar in design to the project settings pages, and there is a reason for it – when you ultimately create a new project from your template, you'll be able to see the very same settings in the project management interface. This is how the page looks like when invoked from the template editor:

And this is the same page in the project management interface:

			e memoq
Templa	te everview		×
Tem	plate name: test_template		
9	Name and meta-information	n	
		Project name	{Client}-{Domain}-{Subject}-{Day}-{Month}-{Year}-
	Metadata	Project	(Client)-(Domain)-(Subject)-(DD)-(MH)-(YYYY)-
		Client	N/A
		Domain	N/A
		Subject	N/A
		Description	N/A
	Options	Record version history	Yes
		Publish on a server after creation	No
	Languages	Source language	ang
		Tarpet language(s)	fre, ger
		Treat sublanguages as separate languages in TB lookup	No
		Show terminology hits that have no target term	Yes
* <u>*</u> *	Workflow settings		
đ	Checkout options		
•2	Communication settings		
μ.	Heavy resources		
4	User assignment		
4	Resources		
×	Language-independent sett	ings	
\$	Content source connection		

			o memo	٩
0 V 8 4 6 5		meQ Localization Project templates 14 43-2215 - memoQ online proj	- 10	Ξ×
Open Anset	Choice Proget Handbaur Verw Control Capy + Control Co	Try Ity Care	And Colds Reverse Constants Determined	
	Overview		Tampet Sanguages	ų
Cvension And or remove target inspector, event horizont, nocific deliverat, nocific deliverat, nocific deliverat, nocific	Declaration Project untings Tempelate research text, tempelate (Vie W Name and meta information	er) Project name	memory constration Project Tempidae (1445-2016	
Translations	Minudata	Project Charlit Constin Scient	(Sant)=(Durant)=(Salger)=(D0)=(MH)=(YYYY)= memory Longitudian	
Talary .	Outors	Casaviption Record contine history	n/s Yu	
		Publish on a server after creation	74	
Reports	Campuages	Source language Target language(s)	erg tra, gar	
		Treat sublanguages as separate languages in TB lookup Show terminology hits that have no target term	Tee Yee	
	Workflow settings			
	Checkani options			
	Leavy resources			
	🐳 User assignment			
	nesources			
	Language Independent setting	S		

A surprising amount of time went into designing it, trying to fit it into as little space as possible, while still allowing the PM to quickly find any setting. Looking at this page reminds you of all the things memoQ is capable of – amazing, isn't it?







Naming patterns

Naming patterns

Our language service provider customers often told us that with the large number of projects they are managing, they need some type of naming convention (call it a rule, system or pattern) to navigate the hundreds of projects they are running. Highly multilingual projects (e.g. those with 30 or 50 target languages) also need a system for naming resources, especially bilingual ones like TMs.

We wanted to help project managers reduce the number of clicks. So we created naming conventions for projects and resources. As an added bonus, PMs can automatically create the properly named resources, and add them to their projects. No more creating fifty TMs and adding them one by one – you just create one naming pattern in the template, and memoQ takes care of the rest. Think about it for a moment! A positive side effect is that when you create projects and resources using templates with naming rules, you have a far smaller chance of accidentally mistyping a resource's name – which would mean you cannot find it later.

To set up a naming rule, you need to use placeholders in the name. There is one for the four metadata fields which belong to practically every object in the memoQ universe: Project, Client, Domain and Subject. Plus some obvious metadata you might want to use in your project and resource names, like source language, target languages or dates.



To use these placeholders in names, just click the "Insert placeholder" link below the text entry field when you are setting up a naming rule:

					e me	emo	q
Edit project template					-		×
Vame and languages	Name and l	angu	ages				۲
Specify project name, languages and meta-	Name and meta-in	formatio	n				
information	Project name	(Client)-{Do	omain)-(Subject)-(DD)-(MN	Record document ve	rsion history		
🔯 Online project settings		1	Project	 Store job details in La Description 	nguage Terminal		
	Project	{Client}-{D	Client				
LiveDocs	Domain		Subject				
	Client		Template				
Translation memories	Subject		SrcLangIso2				
Term bases	memoQ servers	creation	SrcLangiso3 TrgLangiso2 TrgLangiso3				
Muses	Source language	English	TrgLangListiso2				
Settings People Automated actions	Target language(s) //frikaans //finkaans //lbanian //lbanian //lbanian (Macedonia) //lbanian (Montenegro //lb	separate k	CMSConnectionName CMSBatchId CMSBatchName YVYY MM DD HIH-NN	Arabic (Lybia) Arabic (Morocco) Arabic (Marocco) Arabic (Qatar) Arabic (Qatar) Arabic (Saudi Arabia) Arabic (Syrta) Arabic (Syrta) Arabic (U A E)	Arabic (Yemen) Aragonese Araneee Araneee Araneee Assamese Asturian Azeri (Cyrilic) Azeri (Latin)	⊻ She	sw all
Preview full list of template setting	32			ОК	Cancel	Hel	p

memoQ offers you the placeholders in a drop-down list. The names are quite selfexplanatory ("template" refers to the template name, target languages can come one by one, or in a list of all target languages for multilingual projects). We don't have a "date" placeholder, as dates are displayed differently based on Windows' regional settings and on countries – which may be a problem in international organizations. You can set up dates in a uniform and unambiguous way in your organization using the year, month and day placeholders.



Text you type between placeholders is kept unchanged in the name of the project or resource. See the following example:

		e memoq
Name and	languages	
Name and meta-i	information	
Project name	{client}-{domain}-{pro	oject}-{year}
	Insert pla	ceholder 💽
Project		
Domain	tales	
Client	Wilhelm and Jacob	
Subject		

If you create a project from it today, its name will be "Wilhelm and Jacob-tales-" followed by your project name and "-2019". You can check this if you click the small switch to the right of "Insert placeholder":

		e memoq
Name and	languages	
Name and meta	information	
Project name	{client}-{domain}-{proje Insert place	ct}-{year}
Project		
Domain	tales	
Client	Wilhelm and Jacob	
Subject		



In the above example, when you click the switch, this happens:



You can see that {project} is still left as a placeholder. You don't need to provide it in the template – you can do so when you create your project. memoQ won't let you create the project until you've provided values for all metadata placeholders in the template (of course, you don't need to enter manually placeholders like date elements). If you save the template we've created so far, and then click "Create local project from template" (or "Create online project from template"), then choose the template, and try to advance by clicking "Finish", you'll see a red exclamation mark next to the Project field. It warns you that you need to provide a value there. If you now type "Cinderella", your project will be created with the name "Wilhelm and Jacob-tales-Cinderella-2019" (and this will be immediately reflected in the dialog, as you can see on the below screen shot):

You may have also noticed that the Client and Domain fields are greyed out, i.e. disabled – these are values you already provided in the template, and as such should not be changed. Required fields (fields referred to in the template) are marked in bold, like the Project field here. The Subject field on the

	e memoq
Project template	[local] My project template (n/a ->+ 🗸
Project name	Wilhelm and Jacob-tales-Cinderella-20
	Custom name
Client	Wilhelm and Jacob
Project	Cinderella 🕢
Subject	
Domain	tales

above screenshot is not bold, so you don't need to fill it in. You may still want to do so, for example, to be able to search for projects based on their subject.





Example: Automating file exports

Example: Automating file exports

SCENARIO

The automated actions in project templates allow you to add automation at certain points (triggers) in the workflows. In this section, we use one particular scenario to show you how this works. You should explore this functionality as there is a lot you can do here. In this scenario, you are a project manager working for an LSP. Your client has an inhouse review process, and would like a two-column RTF file for this review. At the end of the project, they also want a QA report and the translation memories delivered to them.

SOLUTION

Before we introduced project templates, you would have had to manually export everything. For each project you did for this client, you would have done extra work to export the translation memories, QA reports, or other documents. Project templates can do all that automatically for you.

CREATING THE TEMPLATE

Project templates can found in memoQ's Resource console. Start by creating a new online project template. You will then edit this template. In the "Name and languages" tab shown below, we have entered the client name (always "memoQ"), and made the "Project name" a combination of "{Client}-{Project}-{DD}-{MM}-{YYYY}". So, if the project is called "version-9" and the date is 28th April 2019, the full project name will be "memoQ-version-9-28-04-2019".



You have also decided that for this client, the source language will always be English, but you do not know the target languages until the project is created.



You add a LiveDocs corpus:





Working and master TM

In memoQ, you need a primary (working) TM where memoQ stores the segments when you confirm them.

But you may also have a master TM where you store all translations once the project is finished and delivered. Your master TM contains the final and approved translations. If you check out an online project as translator, and the project manager has set the working TM to be the master TM at the same time, you can change this in your local copy of the project. This way, you can confirm translations into your own TM. Nothing changes in the online project, only in your own local copy.

Examples:

Even though you set the master TM as your working TM, you can only write into it if you have direct permissions for it. For example, if you as translator are not a member of any group, you cannot write into a master TM with default permissions. Due to implicit project permissions, PMs have admin rights to the master TM, but everyone else has lookup rights only.

If you change TMs around in your local copy of the project, and then the PM makes changes in the online project, then the next project synchronization overrides your local settings, and applies the PM's latest settings.

memoQ allows you to differentiate between a working and a master TM in your project. In the project templates, configure automated steps for project wrap-up to delete your working TM and also confirm and update segments into your master TM.



In addition to the working and master TMs, you also add a term base, and assign French and German translators to the project.









Configuring the automated actions

Configuring the automated actions

You now add actions so that files will be exported automatically.

When the translator delivers the document, you want to export a two-column RTF document which will be used by the client for their review. Select "Translator delivers document" from the list of available triggers, then click the "+" button on the right of the "Actions added to the selected trigger" column:





A dialog box opens, showing the list of available actions. Click "Export two-column RTF on server", then click the "Add" button:



You can now choose how to export the two-column RTF document. For now, just accept the default settings:





At the end of the process, you want to automatically export the final document. Choose "All documents complete workflow" from the list of triggers, and add the "Export target documents on server" action:



The client also wants the QA report and the translation memories. To export the QA report, choose the trigger "After project wrap-up", and add "Run QA and export report on server":





Stick with the default folder names here, too:



You can also choose QA and email settings here:

		🧿 memoq
lun QA and create repor	ts	>
Reports and e-mail options	Export QA report on server	
QA options		
Run QA		
If QA is not run, only exis missing	ting QA results will be included in reports, and o we	onsistency check results may be
Send reports in e-mail to		
Translator users	Project managers	
Reviewer 1 users	Reviewer 2 users	
Send to the following add	ditional addresses (separate with semicolon)	
Send separate report	s per translator	



Then choose the trigger "After project wrap-up" again, and add "Export translation memories on server":



Choose to export only the master translation memories:

Export translation memories - settings Export scope Master translation memories All translation memories Export path rules All folder names specified here are relative to the project's file system root. Only the server administrat can change the file system root folder. Translation memores \{TmTrgLangIso2}\{TMName}.{YY}.{MM}.DD HH MM DD HH MM	emoq	ا			
Export scope Master translation memories All translation memories Export path rules All folder names specified here are relative to the project's file system root. Only the server administrat can change the file system root folder. Translation memores \{TmTrgLangIso2}\{TMName}-{YY}-{MM}-{DD}_{{HH}-{NN}} Placeholders (double-click to insert) Placeholders (double-click to insert) Two-letter language code (plus country of present) of the translation memory's target anguage. Two-letter language code (plus country of present) of the translation memory's target anguage.	>			ettings	translation memories - setti
Master translation memories All translation memories Export path rules All folder names specified here are relative to the project's file system root. Only the server administrat can change the file system root folder. Translation memores \{TmTrgLangIso2}\{TMName}-{YY}-{MM}-{DD}_{HH}-{NN} Placeholders (double-click to insert) Description of placeholder Two-letter language code (plus country of present) of the translation memory's target language. YYYY MM DD HH MM M DD HH MM M DD HH MM TmTrgLangle Y					t scope
All translation memories Export path rules All folder names specified here are relative to the project's file system root. Only the server administratic can change the file system root folder. Translation memories \{TmTrgLangIso2}\\{TMName}-{YY}-{MM}-{DD}_{(HH)-{NN}} Placeholders (double-click to insert) Placeholder Two-letter language code (plus country or present) of the translation memory's target language. YYY MMM DD HH H M					aster translation memories
All translation memones Export path rules All folder names specified here are relative to the project's file system root. Only the server administrat can change the file system root folder. Translation memones \{TmTrgLangIso2}\{\TMName}-{YY}-{MM}-{DD}_{(HH}-{NN}) Placeholders (double-click to insert) Two-letter language code (plus country of present) of the translation memory's target language. YY MM DD HH H N N N N N N N N N N N					
Export path rules All folder names specified here are relative to the project's file system root. Only the server administration can change the file system root folder. Translation memones \(TmTrgLangIso2)\(TMName)-{YY}-{MM}-{DD}_{(HH)-{NN}} Placeholders (double-click to insert) Placeholders (double-click to insert) Description of placeholder SrcLangIso3 TmTrgLangIso2 TmTrgLangIso3 YYYY YY MM DD D HH N N					translation memories
All folder names specified here are relative to the project's file system root. Only the server administration can change the file system root folder. Translation memones \[TmTrgLangIso2]\{TMName}-{YY}-{MM}-{DD}_{HH}-{NN} Placeholders (double-click to insert) Placeholders (double-click to insert) Compresent of placeholder Two-letter language code (plus country of present) of the translation memory's target language. YY MM DD HH NM					t path rules
SrcLanglso3 Two-letter language code (plus country of present) of the translation memory's target language. YYYY MM DD HH NM		ourse ourse	opene a ne apatem root. Only the a	i i i i i i i i i i i i i i i i i i i	have a the file system on at folder
Sinc Langlso3 Two-letter language code (plus country of present) of the translation memory's target language. YYYY MM DD HH Nim	.tmx		-{YY}-{MM}-{DD}_{HH}-{NN}	lder. glso2)[\{TMName	ation memores (TmTrgLangiso
TimTrgLanglso3 YYYYY YY MM DD HH	.tmx	er	-{YY}-{MM}-{DD}_{HH}-{NN} Description of placeholder	lder. glso2}[\{TMName t)	ation memories \{TmTrgLanglso,
YYYYY YY MM DD HH	.tmx	er Je (plus cou	-{YY}-{MM}-{DD}_{HH}-{NN} Description of placeholder Two-letter language code	lder. glso2}[{TMName; t)	hange the file system root folder ation memories \{TmTrgLanglso, holders (double-click to insert) inglso3
MM DD HH V	.tmx	er Je (plus cou on memory's	-{YY}-{MM}-{DD}_{HH}-{NN} Description of placeholder Two-letter language code present) of the translation language.	ider. glso2}{{TMName t)	hange the file system root folder ation memories \{TmTrgLanglso, holders (double-click to insert) inglso3 gLanglso2 gLanglso3
DD HH ×	tmx code if rget	er Je (plus cou on memory's	-{YY}-{MM}-{DD}_{HH}-{NN} Description of placeholder Two-letter language code present) of the translation language.	lder. glso2}{{TMName t)	hange the file system root folder ation memories \{TmTrgLanglso, holders (double-click to insert) inglso3 gLanglso2 gLanglso3
HH v	tmx	er Je (plus cou on memory's	-{YY}-{MM}-{DD}_{HH}-{NN} Description of placeholder Two-letter language code present) of the translation language.	Ider. glso2}{{TMName t)	hange the file system root folder ation memories \{TmTrgLanglso, holders (double-click to insert) inglso3 gLanglso3 gLanglso3
NIN	tmx code if rget	er Je (plus cou on memory's	-{YY}-{MM}-{DD}_{HH}-{NN} Description of placeholder Two-letter language code present) of the translation language.	Ider. glso2}[{TMName t)	hange the file system root folder ation memories \{TmTrgLanglso, holders (double-click to insert) inglso3 gLanglso2 gLanglso3
	, tmx	er Je (plus cou on memory's	-{YY}-{MM}-{DD}_{HH}-{NN} Description of placeholder Two-letter language code present) of the translation language.	Ider. glso2)[{TMName t)	hange the file system root folder ation memories \{TmTrgLangIso, holders (double-click to insert) ingIso3 gLangIso2 gLangIso3
QK Cancel H	y code il rget	er Je (plus cou on memory's	-{YY}-{MM}-{DD}_{HH}-{NN} Description of placeholder Two-letter language code present) of the translation language.	ider. glso2)[{TMName rt)	ation memones \{TmTrgLanglso, holders (double-click to insert) inglso3 gLanglso3







Creating a project from the template

Creating a project from the template

Now the template is ready for use. On the "memoQ" tab, click "New Project", then "New online project from template". Or: Under "Recently used templates", click the template name (if it is there).

Choose the content source ("Local", "Content connector", or "CMS"). You can add the files now, or leave that step for later. Then click "Next". In this dialog, the name of the client is already there. Add the name of the project and the target languages. Then click "Finish".



			<u> </u>	memoq
reate new project fro	om template - c-3po.kilgray.com			>
Project template i Select the tem in template pa	information aplate and fill arameters		C	memod
Project template	[c-3po kilgray.com] test_template (\vee	Source language	English	~
Project name	memoQ-eBook-Automation-27-03-2015	Target languages		
	Custom name	French		
lient	memoQ	Geman		
Project	PRJ2705			
Subject	Automation			
Domain	eBook			
Jeadine	2019. 03. 27. 15:32			
	🚹 How should I use these fields?			Change
	Store job details in Language Terminal			
Jescription				



The project opens. Add the files if you haven't done so earlier. Assign the translators and reviewers. They can then begin working on the project.

After the translators have delivered their work, memoQ automatically exports a two-column RTF file which can be sent to the client for review.

Changes made in the client review are then entered into the project files, and the review is finished.

At this stage, memoQ exports the translated target documents. After this, open the project, and on the "Translations" tab, click "Wrap up". The project is now complete, the translation memories and QA reports are exported. On the right hand you can see the "Files" tab with all the automatically exported documents.







About memoQ

memoQ Translation Technologies is the developer of memoQ, one of the world's most advanced translation environments. Used by hundreds of enterprises and translation companies all over the world, memoQ is the #1 solution to automate and optimize the entire localization process, and manage translation and localization projects in a time and cost-efficient manner.

Contact us

Do you have any other questions or concerns regarding a CAT tool, a TMS, integrations or else?

Get in touch with us at sales@memoq.com

