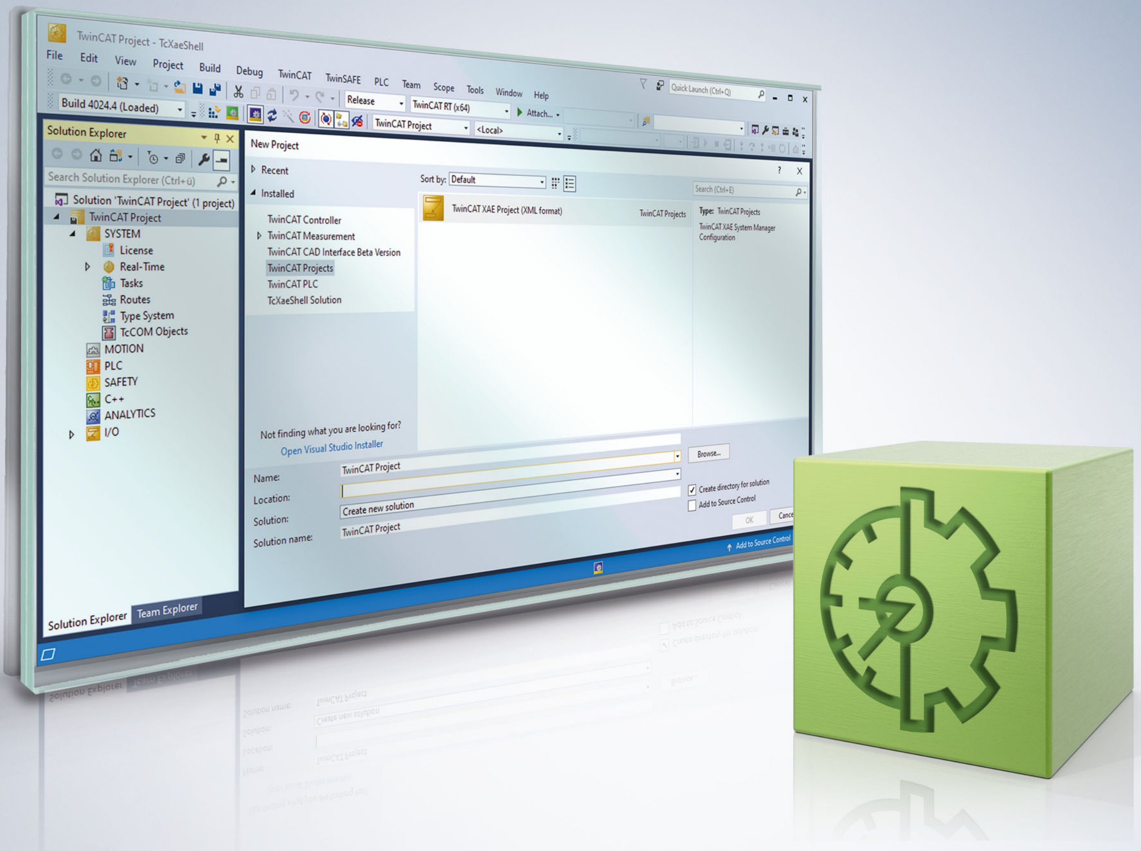
BECKHOFF New Automation Technology

Manual | EN

TE1000

TwinCAT 3 | Source Control



2024-01-23 | Version: 1.4.0

Table of Contents

1 Foreword

1.1 Notes on the documentation

This description is intended exclusively for trained specialists in control and automation technology who are familiar with the applicable national standards.
For installation and commissioning of the components, it is absolutely necessary to observe the documentation and the following notes and explanations.
The qualified personnel is obliged to always use the currently valid documentation.

The responsible staff must ensure that the application or use of the products described satisfies all requirements for safety, including all the relevant laws, regulations, guidelines, and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without notice.
No claims to modify products that have already been supplied may be made on the basis of the data, diagrams, and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered and licensed trademarks of Beckhoff Automation GmbH.
If third parties make use of designations or trademarks used in this publication for their own purposes, this could infringe upon the rights of the owners of the said designations.

Patents

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
and similar applications and registrations in several other countries.

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

Copyright

1.2 For your safety

Safety regulations

Read the following explanations for your safety.
Always observe and follow product-specific safety instructions, which you may find at the appropriate places in this document.

Exclusion of liability

All the components are supplied in particular hardware and software configurations which are appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation, and drive technology who are familiar with the applicable national standards.

Signal words

The signal words used in the documentation are classified below. In order to prevent injury and damage to persons and property, read and follow the safety and warning notices.

Personal injury warnings

⚠ DANGER

Hazard with high risk of death or serious injury.

⚠ WARNING

Hazard with medium risk of death or serious injury.

⚠ CAUTION

There is a low-risk hazard that could result in medium or minor injury.

Warning of damage to property or environment

NOTICE

The environment, equipment, or data may be damaged.

Information on handling the product

This information includes, for example:
recommendations for action, assistance or further information on the product.

1.3 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our https://www.beckhoff.com/secguide.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at https://www.beckhoff.com/secinfo.

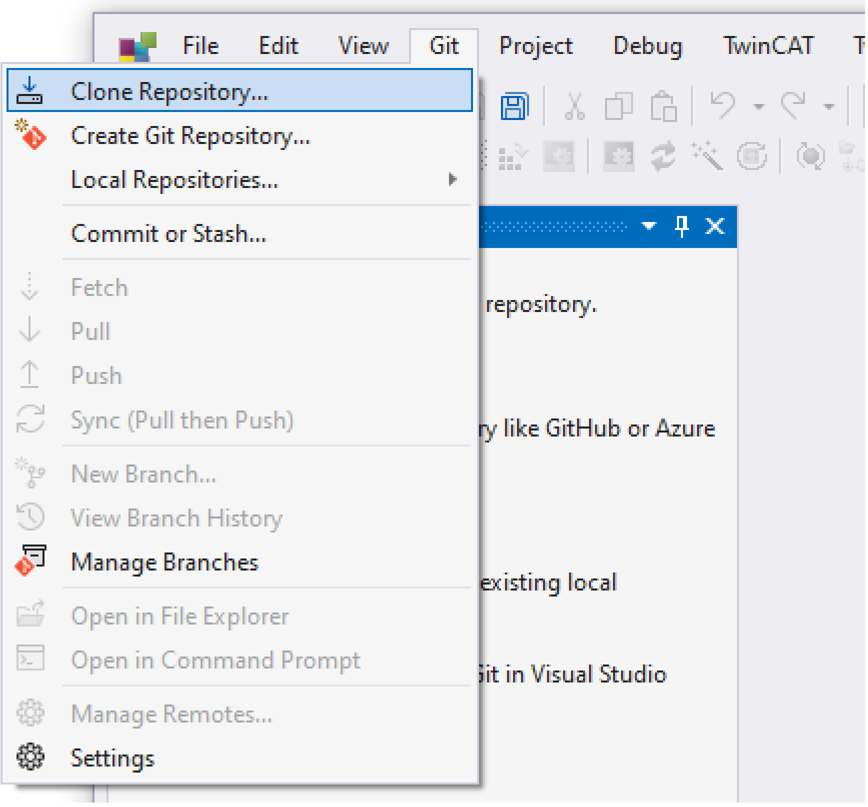
2 Quick start

The following chapter only explains operation from TwinCAT 3.1 Build 4026 onwards. The functions themselves are also available in older TwinCAT versions.

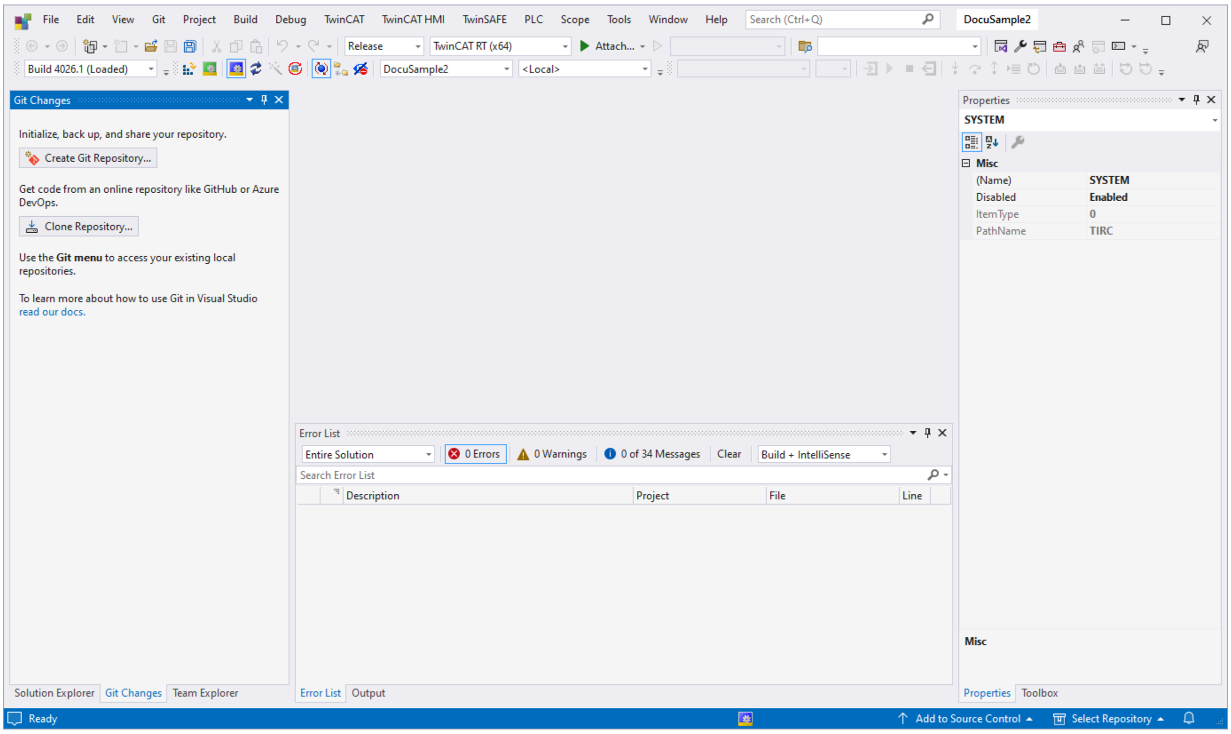
The following chapter explains how to create new projects and clone existing ones. The following quick start guide is based on XAE Shell 64 or Visual Studio 2022 and using Git as the source control client.

Setting up a project with a project that already exists on the server

1. Select Git as the source control client. See Configuration of the Source Control Client [▶ 17].
2. Select the option **Git -> Clone Repository...**.



3. If you have reopened this instance of the XAE Shell, you can also go to or open the **Git Changes** window (**View -> Git Changes**) and select the **Clone Repository...** option.



Initialize, back up, and share your repository.

Get code from an online repository like GitHub or Azure DevOps.

Use the **Git menu** to access your existing local repositories.

To learn more about how to use Git in Visual Studio read our docs.

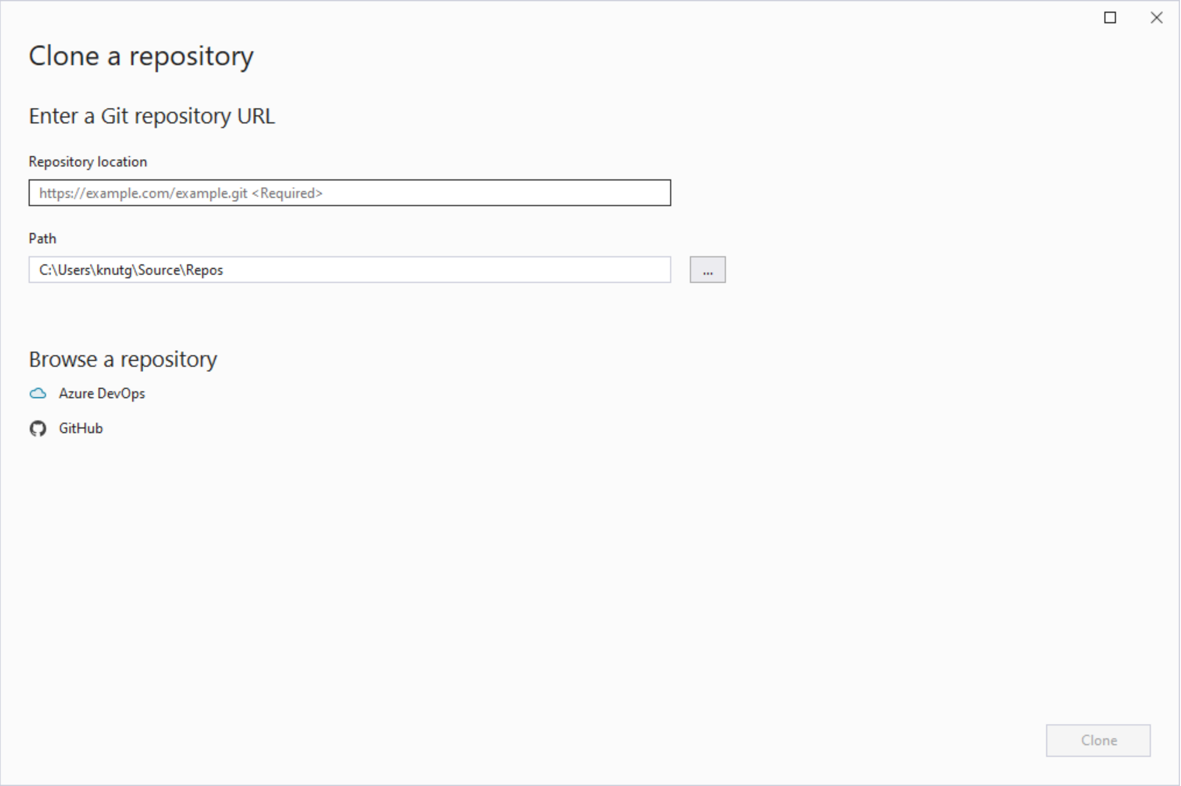
SYSTEM

Misc

(Name)	SYSTEM
Disabled	Enabled
ItemType	0
PathName	TIRC

Misc

⇨ The following window opens:



Clone a repository

Enter a Git repository URL

Repository location: https://example.com/example.git <Required>

Path: C:\Users\knutg\Source\Repos

Browse a repository

4. Enter the path to the Git repository on the server being cloned here, under **Repository location**.
5. Under **Path**, select the folder on your local system that the project will be stored and edited in.
6. Select **Clone** to start cloning.

⇨ After cloning the project, it is automatically opened and the Source Control Management is active.

If there are no global Git settings that define the TwinCAT Project Compare Tool as the default compare and merge tool for TwinCAT files on your system yet, set them before you start working on the project. See Selection of the Compare Tool [▶ 17].

Setting up a project that is not yet under Source Control Management

The procedure depends on the Git server that is to be used. For example, while AzurDevops allows you to also push a local repository if a remote does not already exist on the server, most other servers require you to create a remote repository on a Git server first.

1. Create a Git repository on the remote server to be used.

⇨ The repository exists on the server, but is still empty.

You now have the following two options:

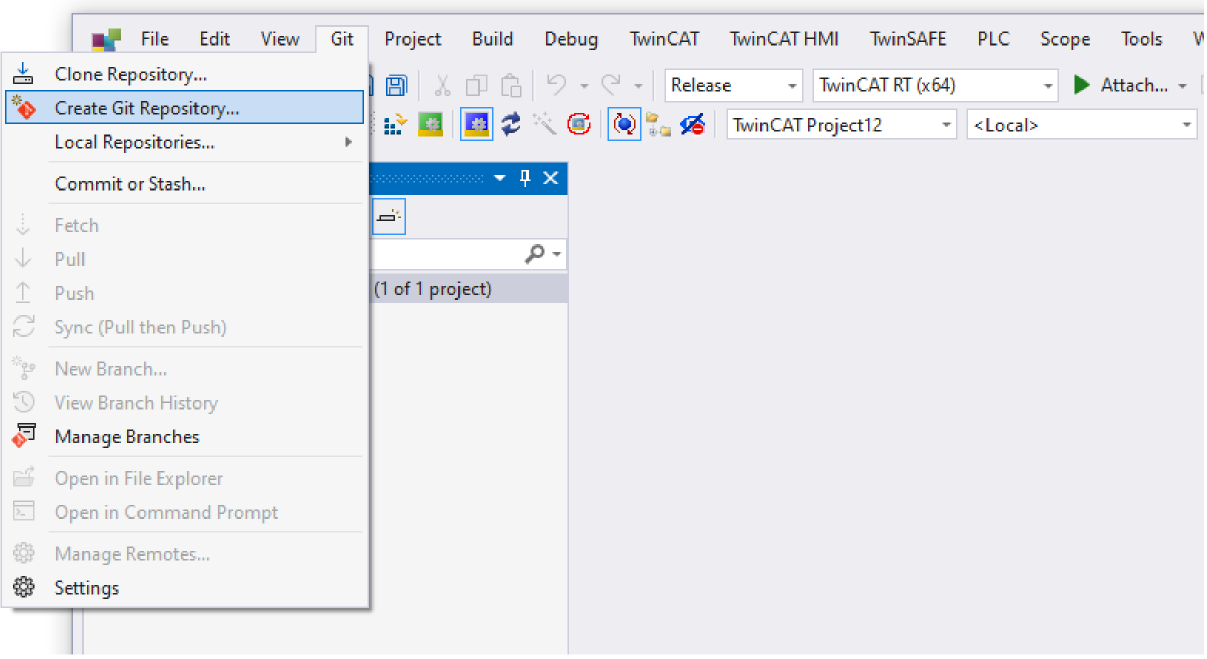
You don't have an existing project and are starting from scratch:

1. Clone the empty repository to the desired working directory (see paragraph Quick start [▶ 8]) and then paste the desired .gitignore file (see .gitignore [▶ 25]).
2. If not already done on your system, configure the TwinCAT Project Compare Tool as default compare/merge tool. See Selection of the Compare Tool [▶ 17].

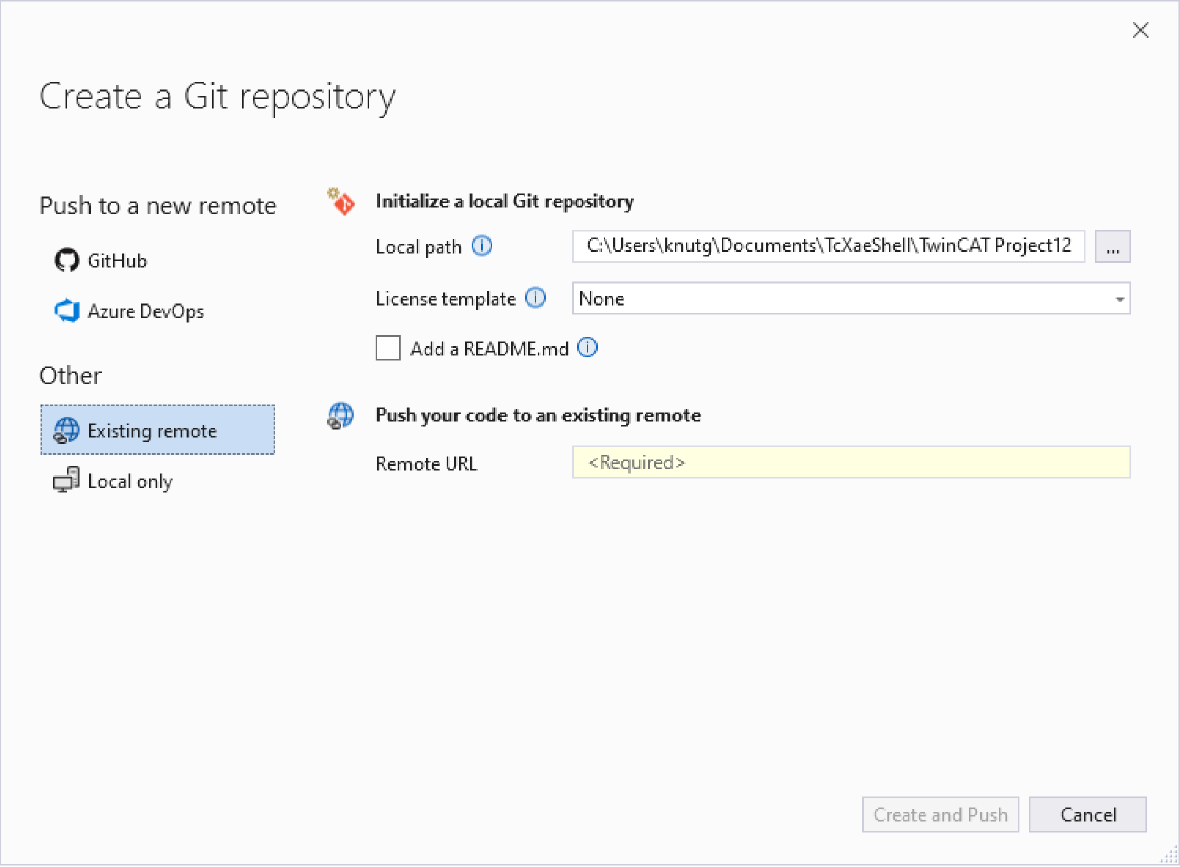
⇨ Afterwards you create a new TwinCAT project in the working directory and start working.

You have an existing project that is not yet under Source Control Management and you just want to push it to a Git server.

1. Store the .gitignore file that will be used in the top project directory. See .gitignore [▶ 25].
2. Create a local Git repository in your working directory. To do this, use the option **Git -> Create Git Repository...**.



⇨ The following window opens:



Create a Git repository

Local path: C:\Users\knutg\Documents\TcXaeShell\TwinCAT Project12

License template: None

□ Add a README.md

Remote URL: <Required>

3. In the **Remote URL** text box, specify the path to the Git repository on the remote server
4. Press the button **Create and Push**.

If you have not placed a .gitignore file in the top project directory, you will get another entry in the dialog shown above, where you can select a gitignore template that is deposited on Github. Since none of the templates on Github are sufficient for a TwinCAT project, deposit the .gitignore file before this step. Otherwise, all data will be transferred to the remote repository and must be removed manually afterwards.

Also see about this

- Configuration of the Source Control Client [▸ 17]
- Selection of the Compare Tool [▸ 17]
- Quick start [▸ 8]
- .gitignore [▸ 25]

3 Integration of the Source Control Management

TwinCAT 3 uses the Visual Studio Framework as basis of the workbench. One of the advantages of this concept is the simple access to the large selection of programming functions in Visual Studio.

Accordingly, TwinCAT also profits from the Source Control Management in Visual Studio.

This section describes how you can use this feature in TwinCAT 3.

3.1 Project Files

So that a TwinCAT project is compatible with Source Control Management systems and supports work in teams, it is stored in several files. The following file extensions exist in the TwinCAT project:

File extension	Source Control Management	Merging permissible	Description
*.tsproj	Yes	Yes, use TwinCAT Project Compare Tool.	TwinCAT project file
*.plcproj	Yes	Yes, use TwinCAT Project Compare Tool.	TwinCAT PLC project file
*.tmc	Yes	Not permissible for PLC projects.	TwinCAT module class (description file for a TcCom module)
*.tpr	Yes	Yes	File contains the refactoring information for standalone PLC projects
*.tpy	No	-	This file serves only for compatibility with programs from other vendors.
*.xti	Yes	Yes, use TwinCAT Project Compare Tool.	In case of multiple file support, parts of the TwinCAT project file are stored with this extension.
*.TcTTO	Yes	Yes, use TwinCAT Project Compare Tool.	PLC task object
*.TcPOU	Yes	Yes, use TwinCAT Project Compare Tool.	PLC program organization unit (POU)
*.TcDUT	Yes	Yes, use TwinCAT Project Compare Tool	PLC data type
*.TcGVL	Yes	Yes, use TwinCAT Project Compare Tool.	PLC global variable list
*.TcVis	Yes	Not currently supported.	PLC visualization
*.TcVMO	Yes	Not currently supported.	PLC visualization manager
*.TcGTLO	Yes	Yes, use TwinCAT Project Compare Tool.	PLC global texts list
*.sln	No	-	This file is the solution file from Visual Studio (VS). It contains, among other data, a tag indicating the VS version used. If this is checked in, it makes working with different VS versions more difficult.
*.suo	No		This is the user options file for the VS project. It contains information on the selected platform, breakpoints, etc. and is user-specific! It is generated when a project is opened on a computer for the first time.

TMC file is regenerated automatically

The description of the process image of a TcCom module is stored in the *.tmc file. If this is to be available directly after the checkout (fetching) of a project, even if the project had not previously been compiled on one's own computer, then the TMC file must be fetched with it (and thus checked in with the project beforehand). The TMC file is automatically re-generated after compiling a PLC project. It is therefore NOT merged for PLC projects and from TwinCAT 3.1 version 4018 also no longer needs to be under Source Control Management!

Do not merge files manually

In general we do not recommend manual merging of the files of a TwinCAT project.

- Always use the TwinCAT Project Compare Tool to merge the files.

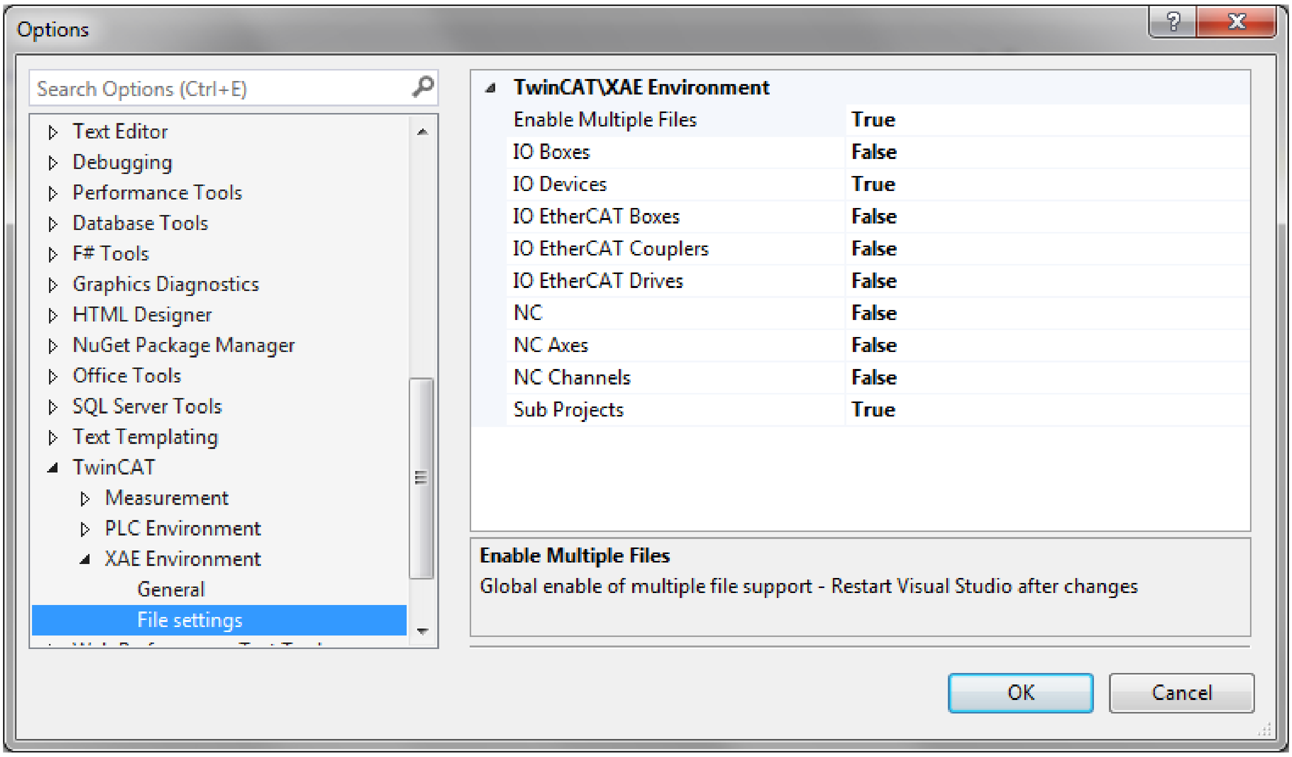
3.2 Project settings

So that a TwinCAT project that is managed using a source code management system can be processed in a team as independently as possible, various settings can be set. These are described below.

Independent Project File

It is possible in TwinCAT to save individual sub-projects, NC axes or I/O devices explicitly in separate files. These files then contain all the parameters, etc. of the respective TwinCAT object, so that the complete project file does not have to be "checked out" when making changes to this object. The links of a PLC project, for example, are thus no longer stored in the TwinCAT project file, but in the corresponding XTI file of the PLC project.

So that this function can be used in TwinCAT it must be "globally" enabled once. This is done via the *TwinCAT –> XAE Environment –> File settings* category in the TwinCAT Engineering environment options.



TwinCAT\XAE Environment	
Enable Multiple Files	True
IO Boxes	False
IO Devices	True
IO EtherCAT Boxes	False
IO EtherCAT Couplers	False
IO EtherCAT Drives	False
NC	False
NC Axes	False
NC Channels	False
Sub Projects	True

Enable Multiple Files
Global enable of multiple file support - Restart Visual Studio after changes

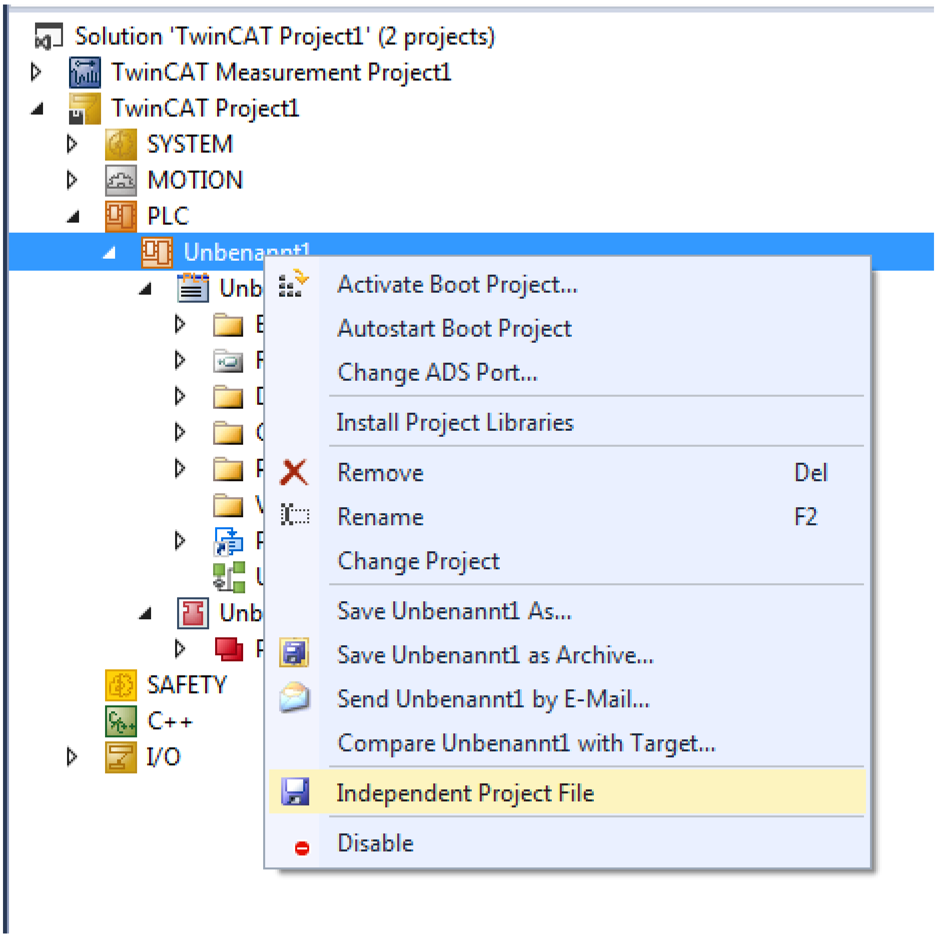
The **Enable Multiple Files** option enables the use of this function for new projects. The options below it define the elements in a new project for which this function is switched on by default.

Restart necessary

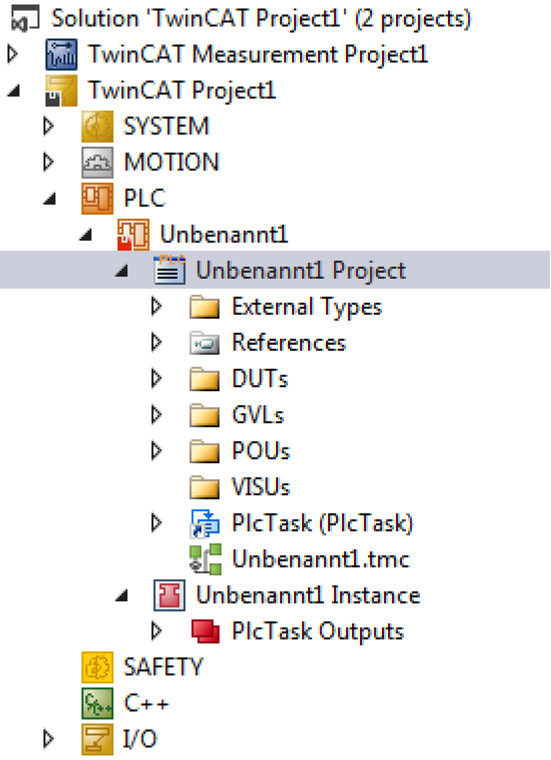
Changes to these options only become effective after restarting the development environment.

Use of Independent Project Files in a project

If this option is not activated by default for a TwinCAT object, it can be done manually (or the option can be reset) via the option **Independent Project File** in the context menu of the TwinCAT object.

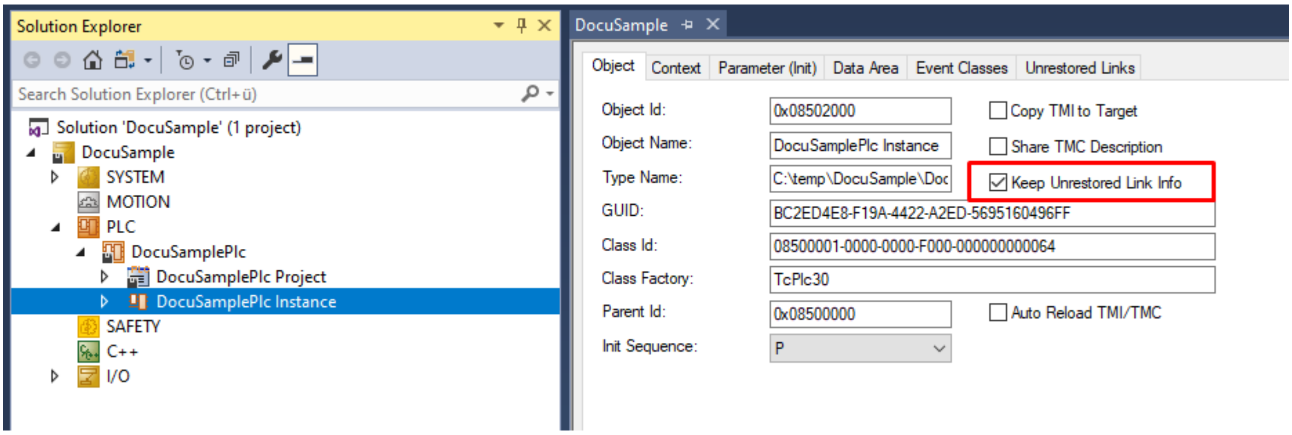


Objects that are saved as **Independent Project Files** are marked in the TwinCAT tree with the aid of an overlay icon in the form of a small floppy disk. A black floppy disk indicates that there are no unsaved changes yet for this object, whereas this is the case with a red floppy disk.



Keep Unrestored Link Info

If not set otherwise, TwinCAT only saves the required information so that the project repository is not unnecessarily loaded. In association with this, TwinCAT also checks whether links are still valid. Any invalid links found are automatically deleted. This mechanism is obstructive when merging projects, since only the code and the links can be merged, but the updated process image is only available after recompiling the code that has now been merged. It is thus possible for the link information to be newer than the process image and the automatic optimization function would delete all links to the new variables in the process image. With the option **Keep Unrestored Link Info**, the link information marked for deletion is retained and automatically restored as soon as the variables show up in the process image.



Object Id: 0x08502000

Object Name: DocuSamplePlc Instance

Type Name: C:\temp\DocuSample\Doc

GUID: BC2ED4E8-F19A-4422-A2ED-5695160496FF

Class Id: 08500001-0000-0000-F000-000000000064

Class Factory: TcPlc30

Parent Id: 0x08500000

Init Sequence: P

□ Copy TMI to Target

□ Share TMC Description

☑ Keep Unrestored Link Info

□ Auto Reload TMI/TMC

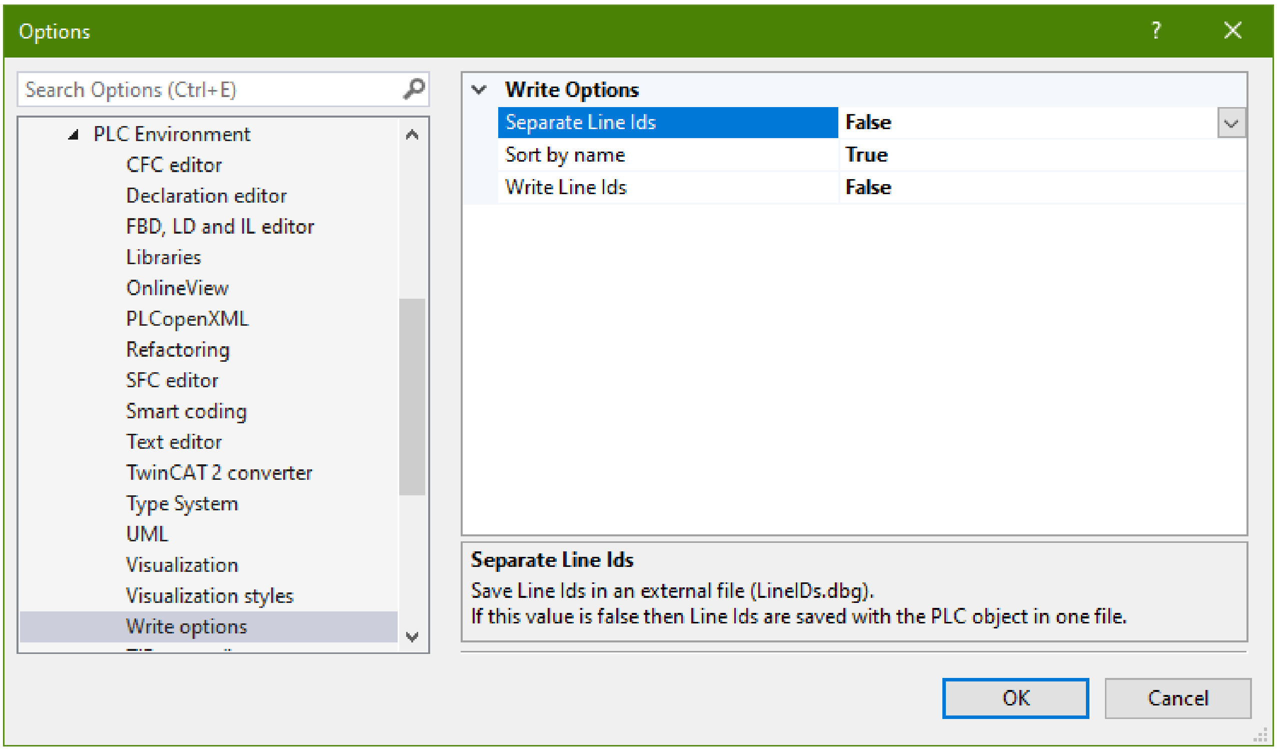
LineIDs

LineIDs are needed to map lines in source code to lines in machine code. This is important as soon as breakpoints are set. Without LineIDs, the following scene would not work as desired:

A breakpoint is set in the source code. After that, blank lines or comments are inserted in the source code before the breakpoint. These are not changes that lead to an online change. The engineering environment is now closed and reopened. Without stored LineIDs it can happen that the set breakpoints point to blank lines. If TwinCAT remembers the LineIDs, the breakpoints point to the desired location.

LineIDs are noted at the end of a POU by default. However, these are changes that are not visible in the PLC editors. Therefore, they are displayed in the Project Compare Tool as formal and not substantive changes. So in case of a merge these changes cannot be merged in a meaningful way. In addition, changes in LineIDs also occur when blank lines and comments are inserted and then deleted. Content-wise, the two POUs are the same in the PLC editors, but not in the XML representation of the file, since the LineIDs may have changed.

To avoid these disadvantages of LineIDs, it is possible to save LineIDs in a separate file (*Separate Line Ids*). They are then no longer part of a POU. Starting with TwinCAT version 3.1.4026 it is also possible to completely disable the saving of LineIDs (*Write Line Ids*). However, this has the disadvantage that all changes in a POU that had no effect on the compilation in the past, e.g. adding blank lines and comments, now require at least one online change.



Write Options

Separate Line Ids	False
Sort by name	True
Write Line Ids	False

Separate Line Ids
Save Line Ids in an external file (LineIDs.dbg).
If this value is false then Line Ids are saved with the PLC object in one file.

3.3 Best Practice

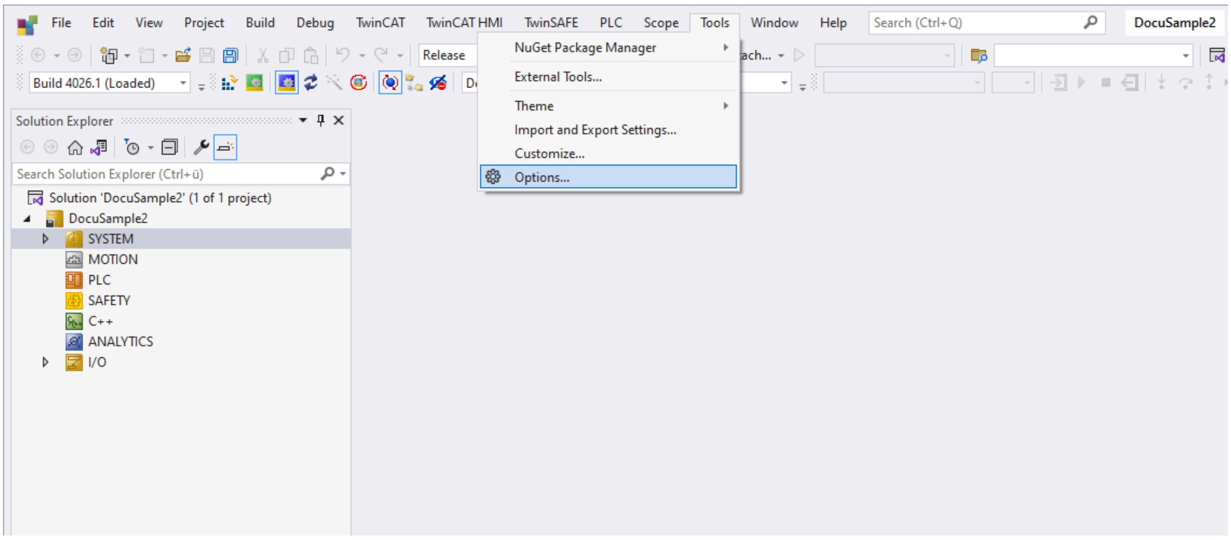
Note the following points when using TwinCAT 3.1 in co-operation with Source Control systems:

- Both the TwinCAT project tree view and the PLC project tree view contain TwinCAT objects whose information is stored in separate files, i.e. not directly in the respective project file. These elements are marked by a small floppy disk in the tree icon. Their contents therefore cannot be accessed in the case of merging the respective project file (files with the extensions *.tsproj and *.plcproj). It is therefore only possible, for example, to add a POU to a PLC project, but the changes inside a POU cannot be accessed. The background to this is that the Source Control Clients do not "look inside" the files that they check out for merging and then also automatically load all the dependencies listed inside them from the Source Control System.
- Since TwinCAT objects in the project files are clearly identified with a GUID, it is **always** necessary to confirm the deletion of an object with a "checkin". For this reason the deletion of an object and the creation of a new object with the same name and the same contents is always a change!
- Some objects in the TwinCAT tree require that they are also loaded in the background in order to display them in the tree. Thus all methods, actions, etc. of a POU are also stored in it. For this reason it is a good idea to close a project in TwinCAT first before fetching it again from the Source Control System.

4 Configuration of the Source Control Client

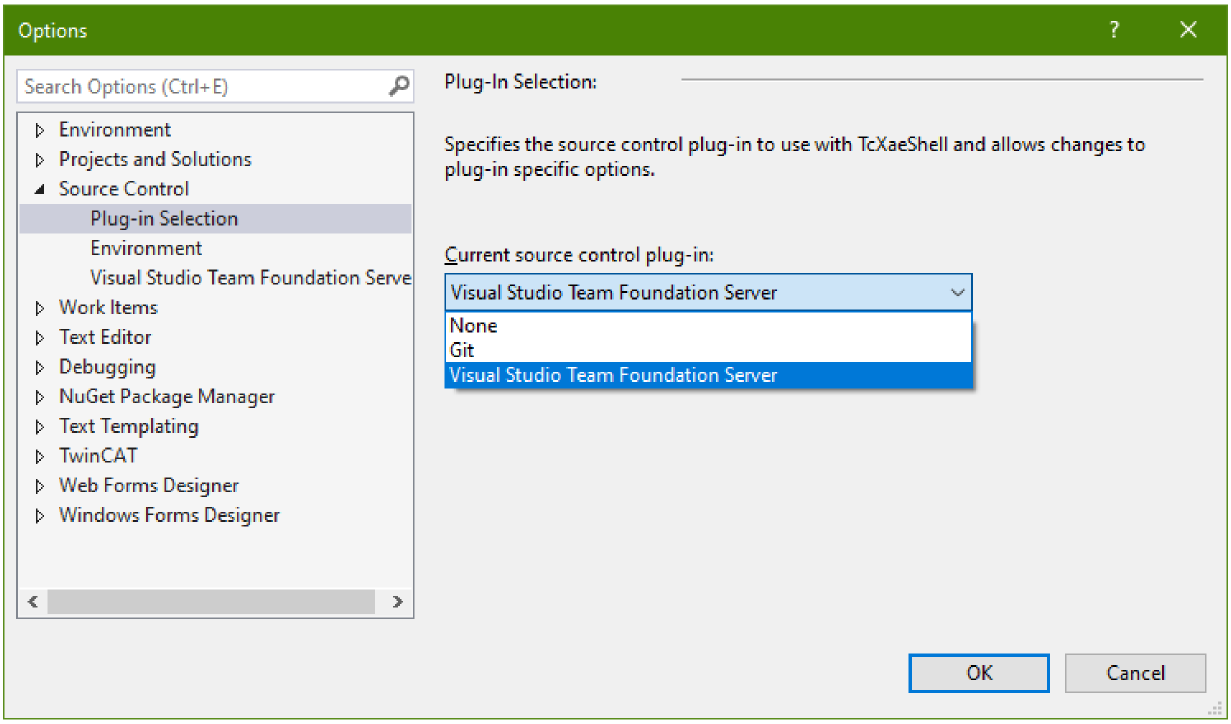
Select the Source Control Client as follows:

1. Select the menu **Tools** -> **Options** of the TcXaeShell.



⇨ The dialog box displays all the settings of the TcXaeShell environment.

2. Go to **Source Control->Plug-in Selection** and select **Visual Studio Team Foundation Server.**



Plug-In Selection:

Specifies the source control plug-in to use with TcXaeShell and allows changes to plug-in specific options.

Current source control plug-in: Visual Studio Team Foundation Server

3. Confirm with **OK**.

Further settings may be required depending on the selected Source Control Management system. They ensure that the Project Compare Tool is called for comparing TwinCAT projects. These settings are generated automatically by the TwinCAT Project Compare Tool for clients of Visual Studio Team Foundation Server and Git.

4.1 Selection of the Compare Tool

Following the selection of the Source Control Client to be used in the project, the client must also be configured so that the TwinCAT Project Compare Tool is used to compare and merge TwinCAT files. The configuration of the individual clients can significantly differ.

Whereas clients such as TFS or PlasticsSCM allow different Compare Tools to be defined for the various types of file, there are also clients that only permit a general Compare Tool to be set (e.g. AnkhSVN). Beyond that, Source Control Systems such as Git also allow a distinction to be made between local project settings and global settings.

For the Source Control Clients used most frequently with TwinCAT (TFS, Git and AnkhSVN), the configuration of the Compare Tool can be automated from the TwinCAT Project Compare Tool.

Since not all features of the different clients can be discussed here, please refer to section Transfer parameters [▶ 23] of the Project Compare Tool for further details.

Define Compare Tools yourself

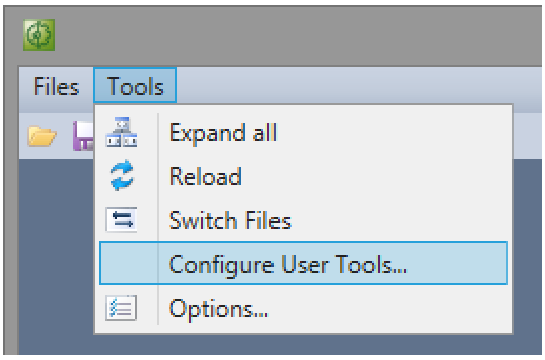
Some clients only allow a general compare/merge tool for all file extensions. For this reason the TwinCAT Project Compare Tool offers the option to define further Compare Tools for special file extensions. See Configuration of the TcProjectCompare for use with Source Control.

Configuration of the TwinCAT Project Compare Tool

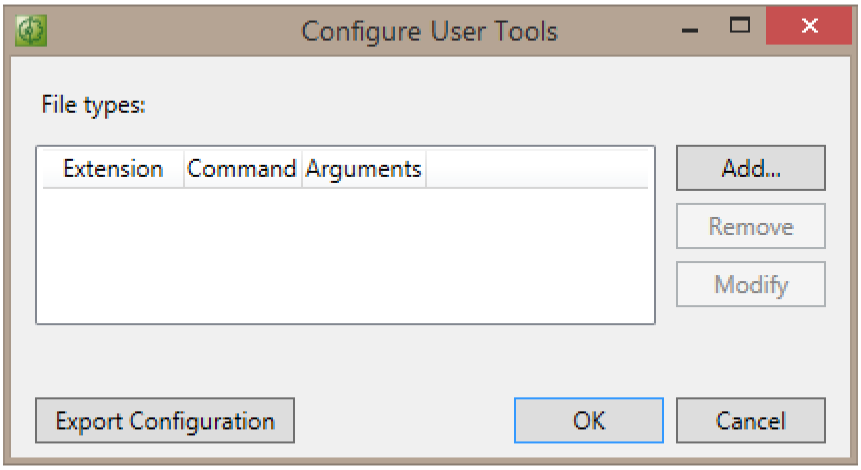
In order to call the TwinCAT Project Compare Tool from the TwinCAT development environment, the selected Source Control Plugins must be configured accordingly in TcXaeShell.

Setting the TwinCAT Project Compare Tool as standard tool for Compare-and-Merge functions for subversion Git or Microsoft Team Foundation Server:

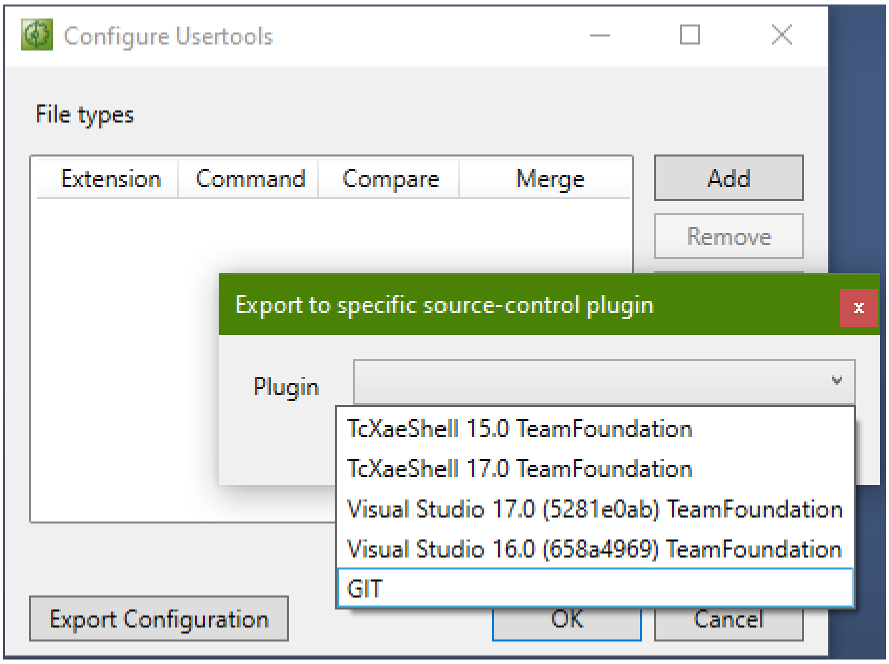
1. Open **TwinCAT-> TcProjectCompare**.
2. Select **Tools -> Configure User Tools....**



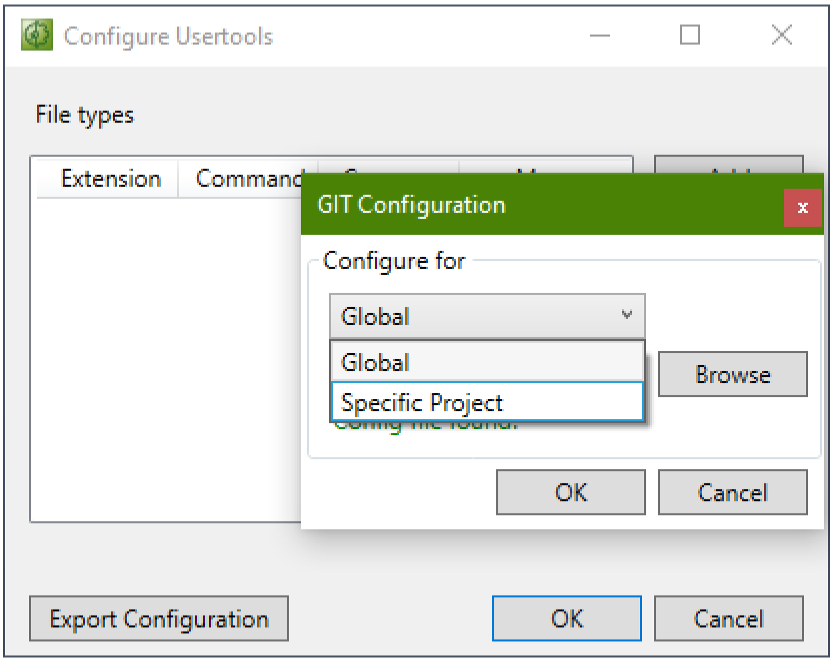
3. In the following dialog, select a default tool for the **Merge**.
   ⇨ This tool is used for all areas outside of the PLC, e.g. for the comparison of specifications for tasks.
4. Click **Export Configuration**.



5. If the required Source Control Plugin has already been selected in TcXaeShell, you can export the specifications for this plugin if you are using Git or Microsoft Team Foundation Server.


Configure Usertools
File types
Extension
Command
Compare
Merge
Add
Remove
Export to specific source-control plugin
x
Plugin
TcXaeShell 15.0 TeamFoundation
TcXaeShell 17.0 TeamFoundation
Visual Studio 17.0 (5281e0ab) TeamFoundation
Visual Studio 16.0 (658a4969) TeamFoundation
GIT
Export Configuration
OK
Cancel

6. If you have selected Git as the Source Control Client, you can decide below whether to update the global or project-specific Config file.



Also see about this

- Configuration of the TcProjectCompare for use with source control [▸ 20]

5 Configuration of the TcProjectCompare for use with source control

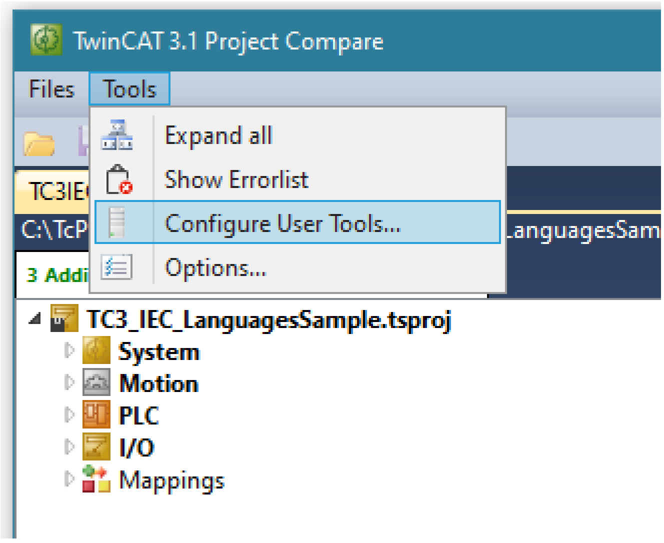
The configuration of the TcProjectCompare can be generated from the software for use with some of the most frequently used source control clients. At the moment, these are the following clients:

- Microsoft Team Foundation Server
- Git
- AnkhSVN

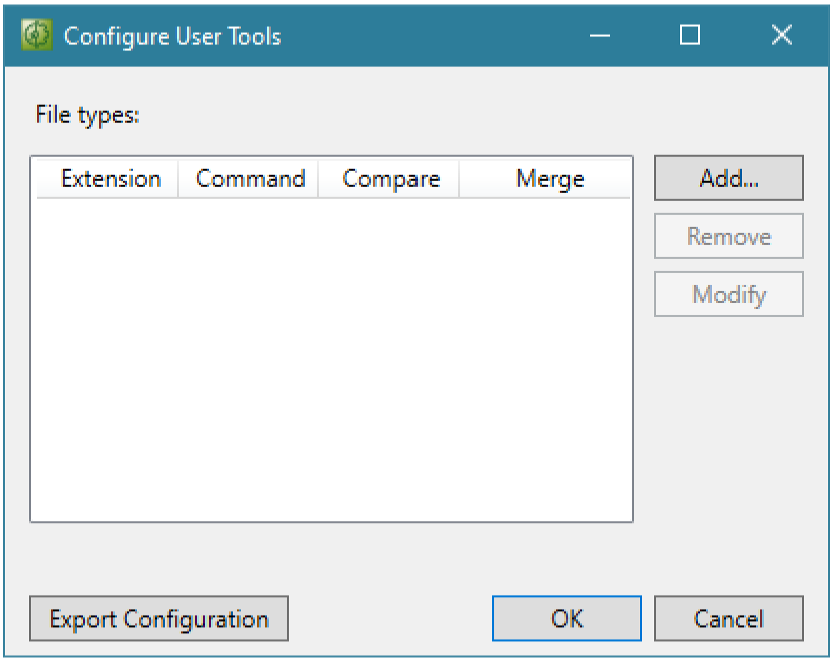
Generating the configuration of these clients

To generate the settings for one of the source control clients listed above, proceed as follows:

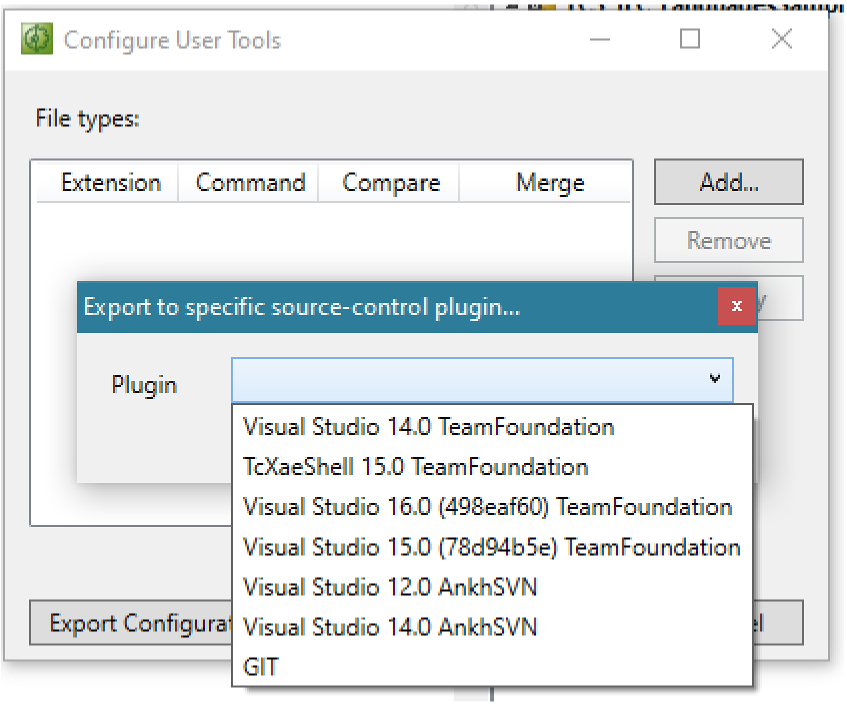
1. Open TcProjectCompare**.**
2. Select the menu **Tools** -> **Configure User Tools....**



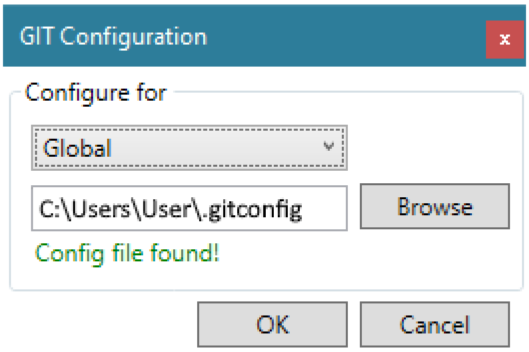
3. Click on **Export Configuration.**



4. In the following dialog, select the source control client for which the settings are to be generated from the selection box.


Configure User Tools
File types:
Extension
Command
Compare
Merge
Add...
Remove
Export Configurat
Export to specific source-control plugin...
Plugin
Visual Studio 14.0 TeamFoundation
TcXaeShell 15.0 TeamFoundation
Visual Studio 16.0 (498eaf60) TeamFoundation
Visual Studio 15.0 (78d94b5e) TeamFoundation
Visual Studio 12.0 AnkhSVN
Visual Studio 14.0 AnkhSVN
GIT

5. In case Git is the source control client, you can select in the following dialog whether you want the settings for the selection of the Compare Tool used to be saved globally or for the specific project. If you save them for the specific project, select the *.gitconfig* file from your local repository. Then confirm your selection with **OK**.


GIT Configuration

Configure for: Global

C:\Users\User\.gitconfig

Config file found!

⇨ The configuration has been created.

Configuring other source control clients:

If you use a source control client that is not listed above, you must configure the use of the TcProjectCompare in the respective source control client. To do this, use the transfer parameters listed in the section Transfer parameters of the TcProjectCompare [▶ 23].

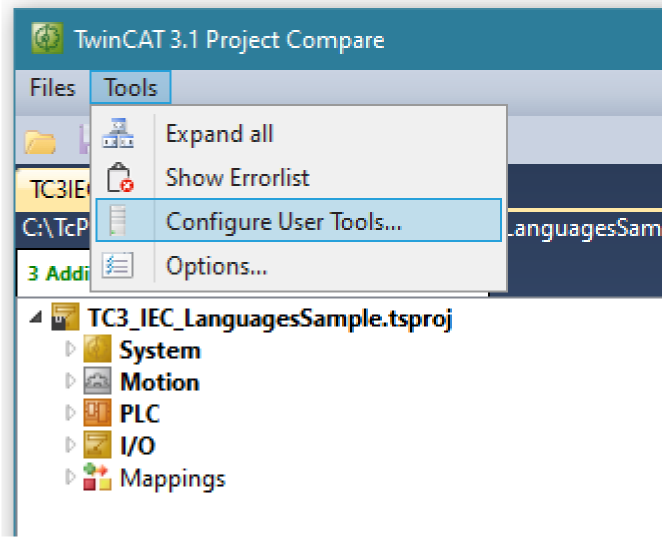
Dealing with non-TwinCAT flies

Some clients only allow a general compare/merge tool for all file extensions. For this reason the TwinCAT Project Compare tool offers the option to define further compare tools for special file extensions.

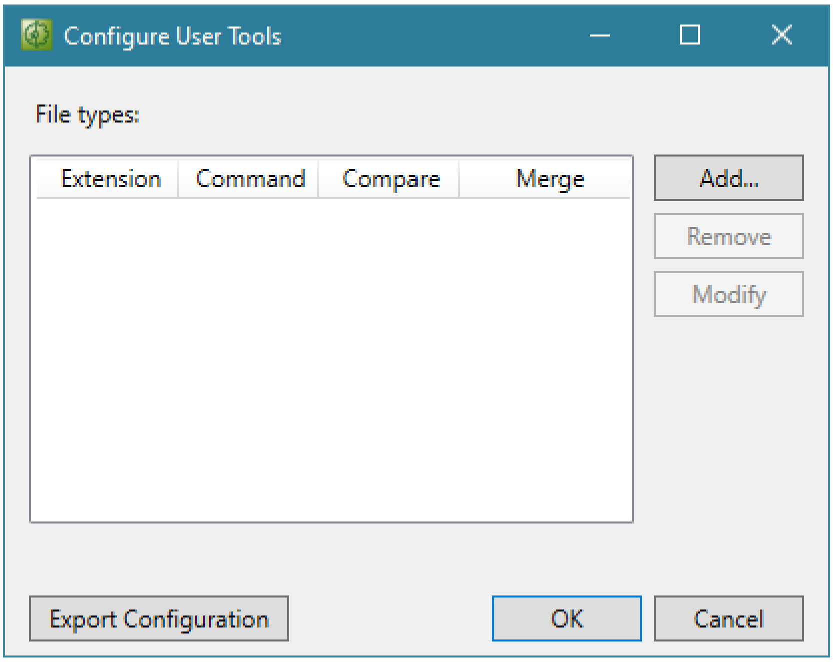
Setting an external compare tool for file extensions:

1. Open the TcProjectCompare**.**

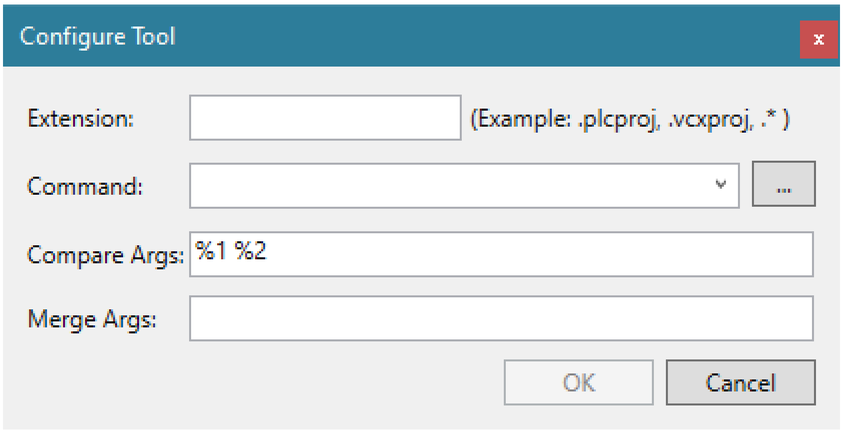
2. Select the menu **Tools** -> **Configure User Tools....**



3. Click on **Add**.



4. In the following dialog describe which file extension is used, which tool is to be called (**Command** line) and which transfer parameters are required for compare or merge. Confirm the settings with **OK**.



Extension: (Example: .plcproj, .vcxproj, .*)

Command:

Compare Args: %1 %2

Merge Args:

5.1 Transfer parameters of the TcProjectCompare

Since not all special features of the various clients can be dealt with here, the transfer parameters of the TwinCAT Project Compare Tools are briefly described below:

Transfer parameters for TwinCAT version 4020 or higher:

/filel	File path – left side.
/filer	File path, right side.
/filem	File path for the merged file.
/dl	Display name, left side.
/dr	Display name, right side.
/sc	Call via source control client.

Transfer parameters for TwinCAT versions 4018 or lower:

	File path – left side.
	File path, right side.
	File path of merged file.
/dl	Display name, left side.
/dr	Display name, right side.
/sc	Call via source control client.

In Project Compare tool version or lower 4018 the file paths for the left, right and merged file were determined based on the order in which they are called. The order was always left, right, merged file.

Samples:

Compare (Compare/Diff):

Version 4018 for PlasticSCM:

```
C:\TwinCAT\3.1\Components\TcProjectCompare\TcProjectCompare.exe
"@destinationfile" "@sourcefile" /sc
```

Version 4018 for TFS:

```
C:\TwinCAT\3.1\Components\TcProjectCompare\TcProjectCompare.exe %2 %1 /dl %7 /dr
%6 /sc
```

Version 4020 for TFS:

```
C:\TwinCAT\3.1\Components\TcProjectCompare\TcProjectCompare.exe /filel %2 /filer
%1 /dl %7 /dr %6 /sc
```

Merge:

Version 4018 for PlasticSCM:

```
C:\TwinCAT\3.1\Components\TcProjectCompare\TcProjectCompare.exe /dl
"@destinationsymbolic" /dr "@sourcesymbolic" "@destinationfile" "@sourcefile"
"@output" /sc
```

Version 4018 for TFS:

```
C:\TwinCAT\3.1\Components\TcProjectCompare\TcProjectCompare.exe %2 %1 %4 /dl
%7 /dr %6 /sc
```

Version 4020 for TFS:

```
C:\TwinCAT\3.1\Components\TcProjectCompare\TcProjectCompare.exe /filel %2 /filer
%1 /filem %4 /dl %7 /dr %6 /sc
```

6 Merging TwinCAT PLC projects

If more than one developer is working on the same PLC project, conflicts may arise when "checking in/committing" the changes. For example, parallel POUs can be added/edited by more than one developer of the PLC project. These conflicts must be solved if a consistent project is to be created. This procedure for combining changes is called merging.

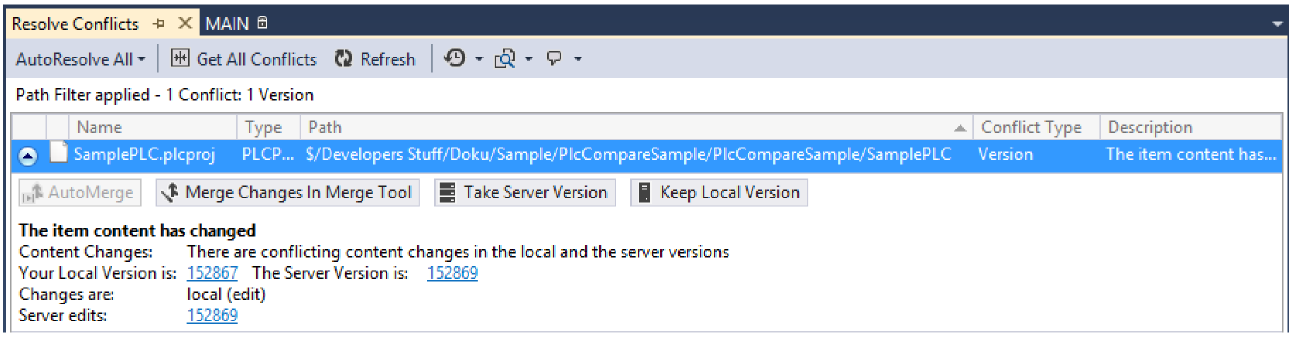
NOTICE

Loss of data

The "automerge" function (automatic merging) of Source Control Clients can lead to the loss of PLC objects (POUs, GVL, etc.).

- Deactivate automatic merging of Source Control Clients
- TwinCAT Project Compare Tool must be configured as Merge Tool. See Configuration of the TcProjectCompare for use with source control [▶ 20].

In case of conflicts when checking in the PLC project, the following dialog appears in the TSF Client:



Path Filter applied - 1 Conflict: 1 Version

Name	Type	Path	Conflict Type	Description
SamplePLC.plcproj	PLCP...	$/Developers Stuff/Doku/Sample/PlcCompareSample/PlcCompareSample/SamplePLC	Version	The item content has...

The item content has changed

Content Changes: There are conflicting content changes in the local and the server versions

Your Local Version is: 152867 The Server Version is: 152869

Changes are: local (edit)

Server edits: 152869

Solve these problems as follows:

1. Select the option **Merge Changes In Merge Tool**.
2. After merging a PLC project (file extension *.PLCproj), request all files that were added to the PLC project with **get latest version**.

7 Appendix

7.1 .gitignore

Example of a *.gitignore*-file for a TwinCAT 3 project:

```
### TwinCAT3 ###
# website: https://www.beckhoff.com/twincat3/

# TwinCAT PLC
*.plcproj.bak
*.plcproj.orig
*.tpy
*.tclrs
*.library
*.compiled-library
*.compileinfo
*.asm
*.core
LineIDs.dbg
LineIDs.dbg.bak

# TwinCAT C++ and shared types
# ignoring the TMC file is only useful for plain PLC programming
# as soon as shared data types (via tmc), C++ or in general TcCom-Module are used, the TMC file has
to be part of the repository
*.tmc
*.tmcRefac

# TwinCAT project files
*.tsproj.bak
*.tsproj.b?k
*.tsproj.orig
*.xti.bak
*.xti.bk?
*.xti.orig
*.xtv
*.xtv.bak
*.xtv.bk?

# Multiuser specific
**/.TcGit/

# exclude not required folders
**/_Boot/
**/_CompileInfo/
**/_Libraries/
**/_ModuleInstall/
**/_Deployment/
**/_Repository/

# VS Shell project specific files and folders
**/.vs/
*.~u
*.project.~u
*.suo
```

8 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Download finder

Our download finder contains all the files that we offer you for downloading. You will find application reports, technical documentation, technical drawings, configuration files and much more.

The downloads are available in various formats.

Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on our internet page: www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline:	+49 5246 963-157
e-mail:	support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline:	+49 5246 963-460
e-mail:	service@beckhoff.com

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone:	+49 5246 963-0
e-mail:	info@beckhoff.com
web:	www.beckhoff.com

More Information:

www.beckhoff.com/te1000

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com