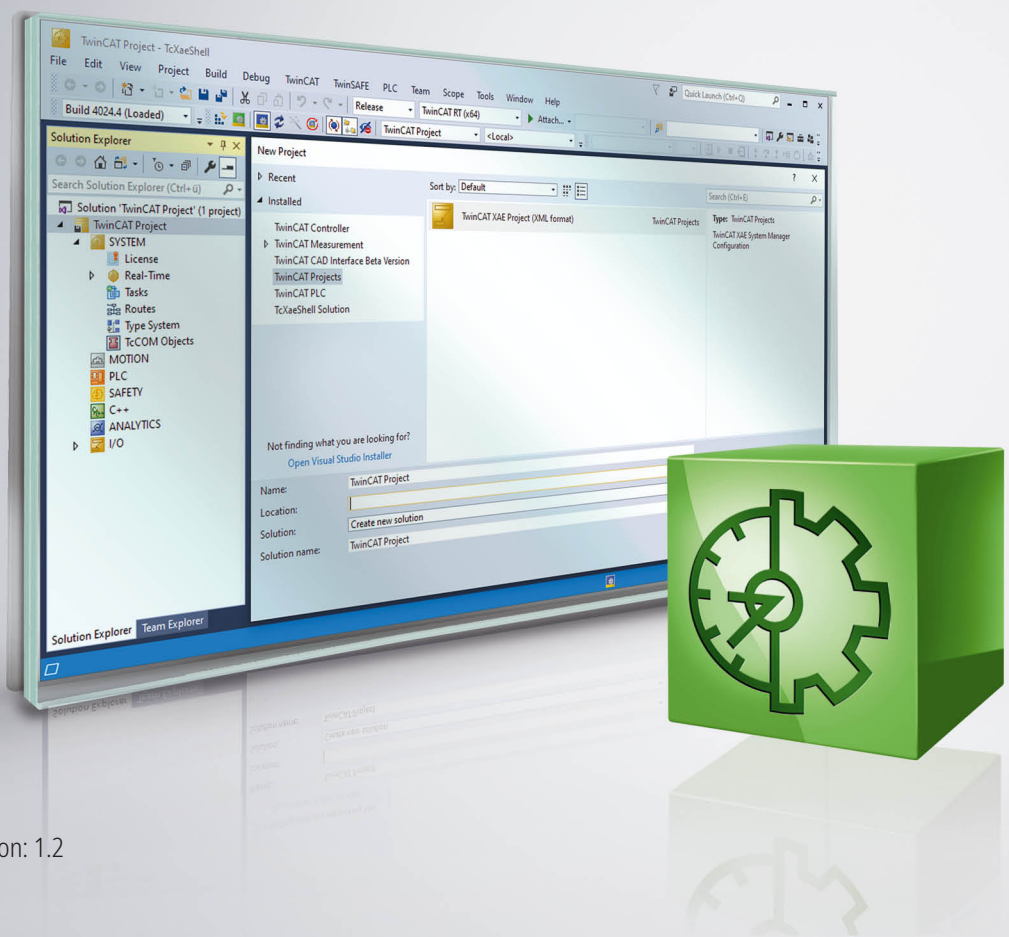
BECKHOFF New Automation Technology

Manual | EN

TE1000

TwinCAT 3 | Source-Control



2021-01-28 | Version: 1.2

Table of Contents

1 Foreword

1.1 Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the 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 prior announcement.
No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.
Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

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

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

Copyright

1.2 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

All the components are supplied in particular hardware and software configurations 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 engineering who are familiar with the applicable national standards.

Description of symbols

In this documentation the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

⚠ DANGER

Serious risk of injury!

Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.

⚠ WARNING

Risk of injury!

Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.

⚠ CAUTION

Personal injuries!

Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.

NOTE

Damage to the environment or devices

Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.

Tip or pointer

This symbol indicates information that contributes to better understanding.

2 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.

2.1 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	TwinCAT project file
*.plcproj	yes	yes, use TwinCAT Project Compare	TwinCAT PLC project file
*.tmc	yes	not permissible for PLC projects	TwinCAT module class (description file for a TcCom module)
*.tpy	no	-	This file serves only for compatibility with programs from other vendors.
*.xti	yes	yes, use TwinCAT Project Compare	In case of multiple file support, parts of the TwinCAT project file are stored with this extension.
*.TcTTO	yes	yes, use TwinCAT Project Compare	PLC task object
*.TcPOU	yes	yes, use TwinCAT Project Compare	PLC program organizational unit (POU)
*.TcDUT	yes	yes, use TwinCAT Project Compare	PLC data type
*.TcGVL	yes	yes, use TwinCAT Project Compare	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	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.

NOTE

TMC file is automatically re-generated

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 translated 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 translating 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 administration!

NOTE

No manual merging of files

Manual merging of the files of a TwinCAT project is not generally recommended. The TwinCAT Compare Tool should be always used for this!

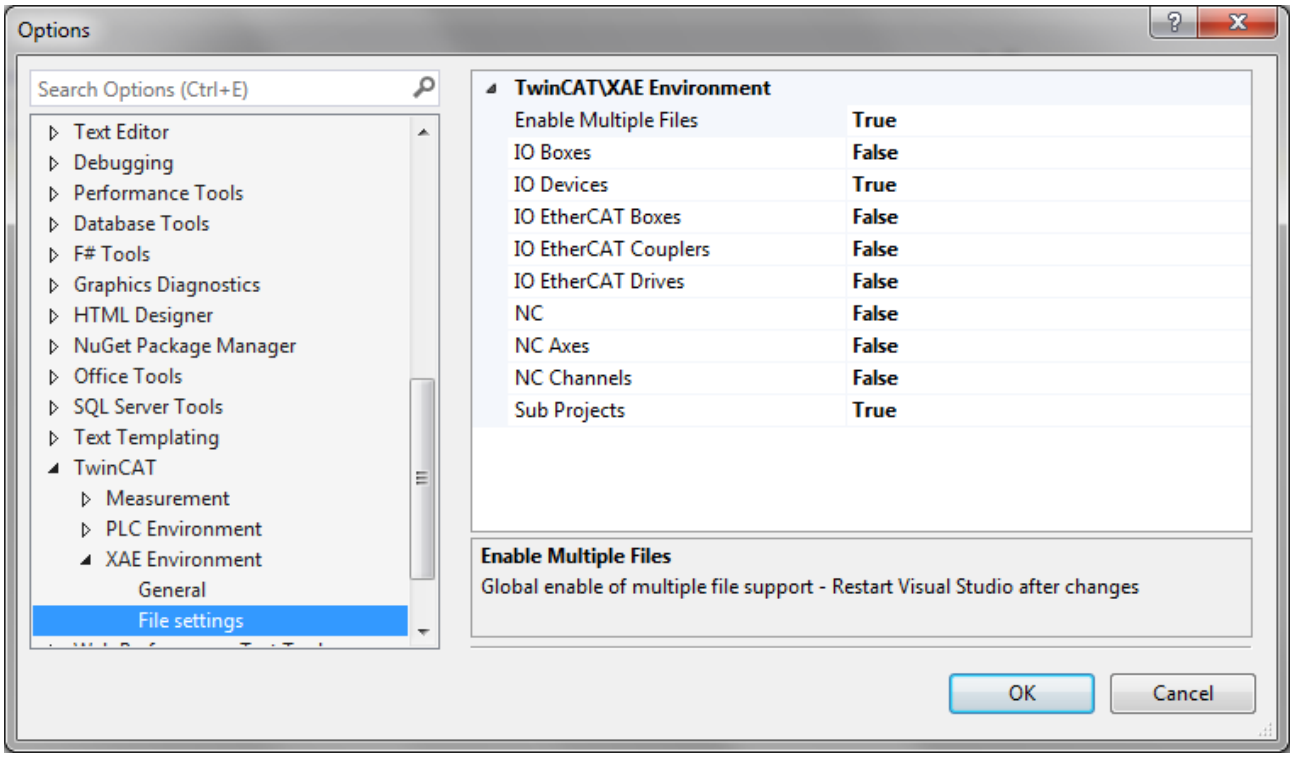
2.2 Project settings

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 IO 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.



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.

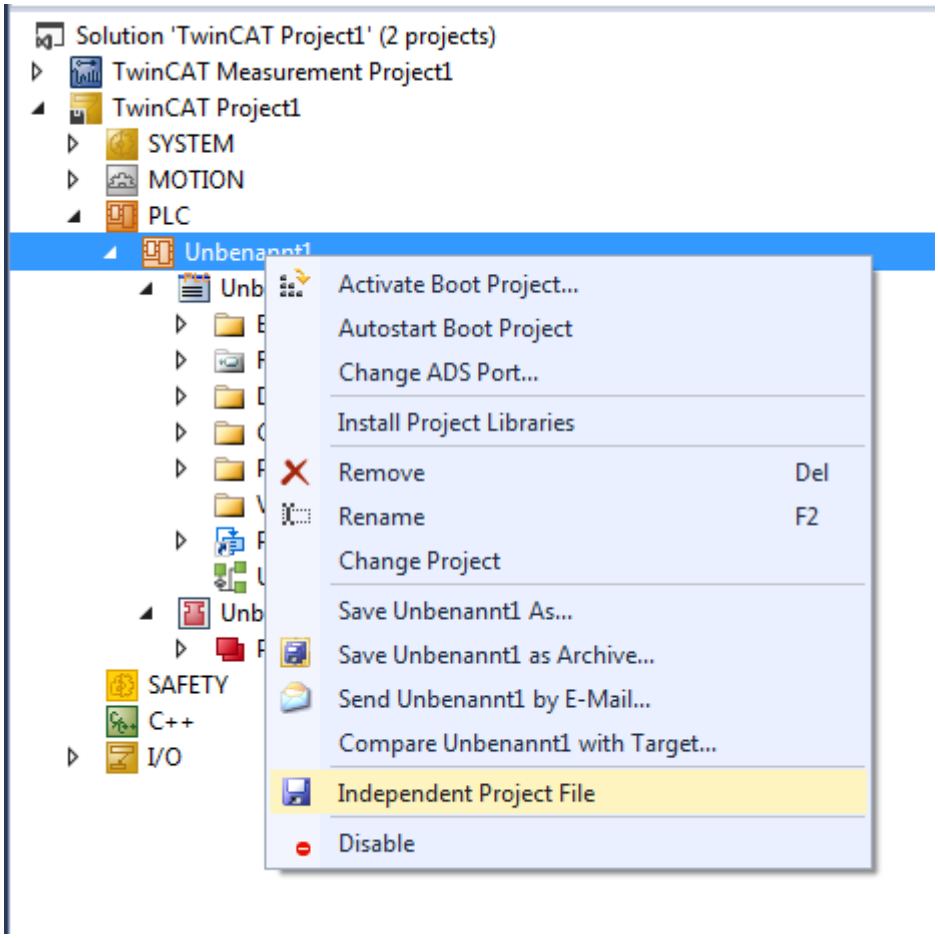
NOTE

Restart required

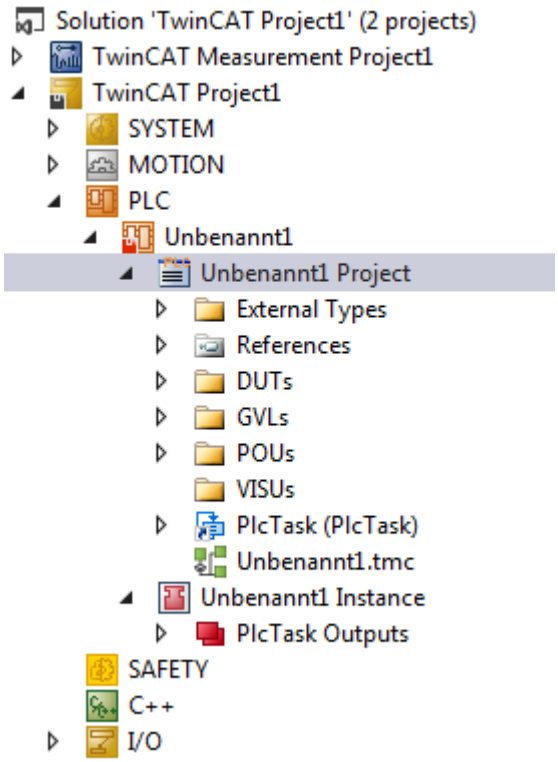
Changes to these options only become effective after restarting the development environment!

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.

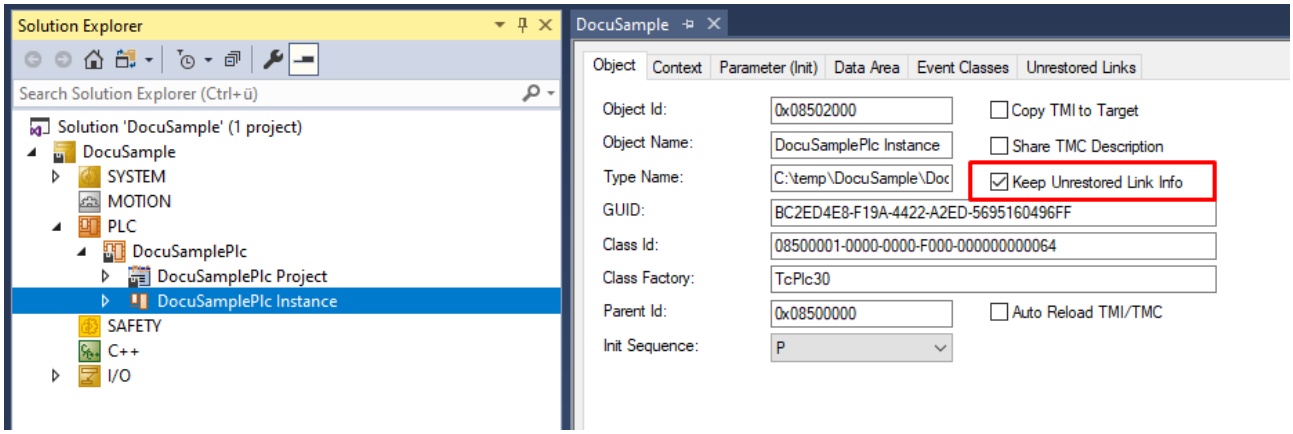


Objects that are saved as Independent Project Files are marked in the TwinCAT tree with the aid of an overlay icon in the form of a small floppy disk. A black floppy disk indicates that there are no unsaved changes yet for this object, whereas this is the case with a red floppy disk.



Keep unrestored Links

Unless set differently, TwinCAT tries to avoid unnecessary loads on the project repository by no longer storing information that is no longer required. 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 links", the link information marked for deletion is retained and automatically restored as soon as the variables show up in the process image.



2.3 Best Practice

The following points should be borne in mind 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

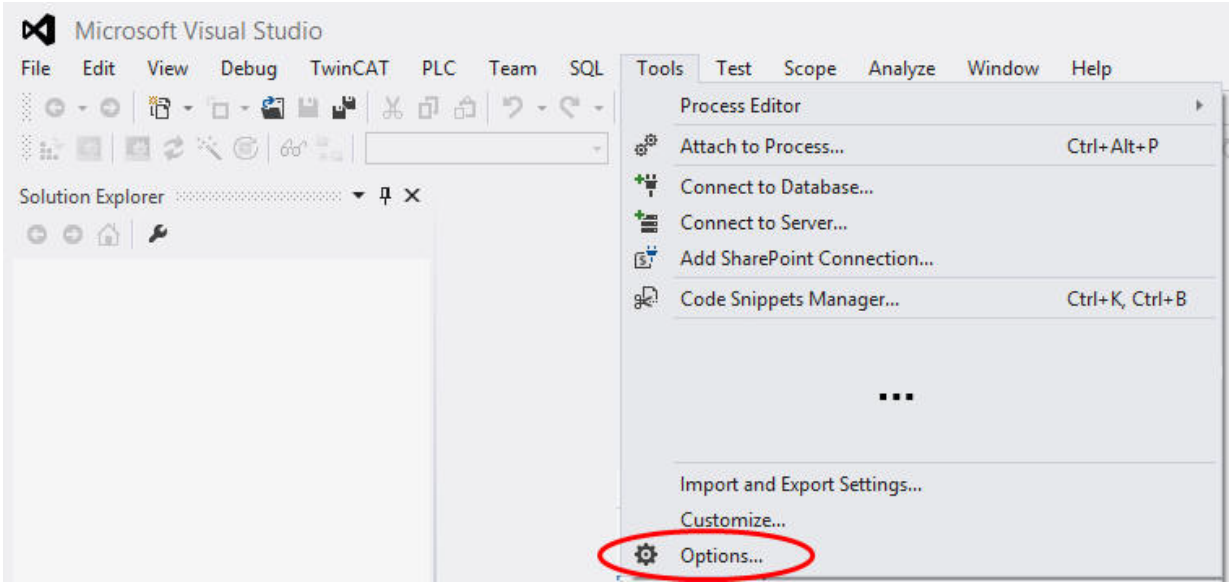
accessed. The background to this is that the Source Control Clients do not “look inside” the files that they check out for merging and then also automatically load all the dependencies listed inside them from the Source Control System.

- 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.

3 Configuration of the source control client

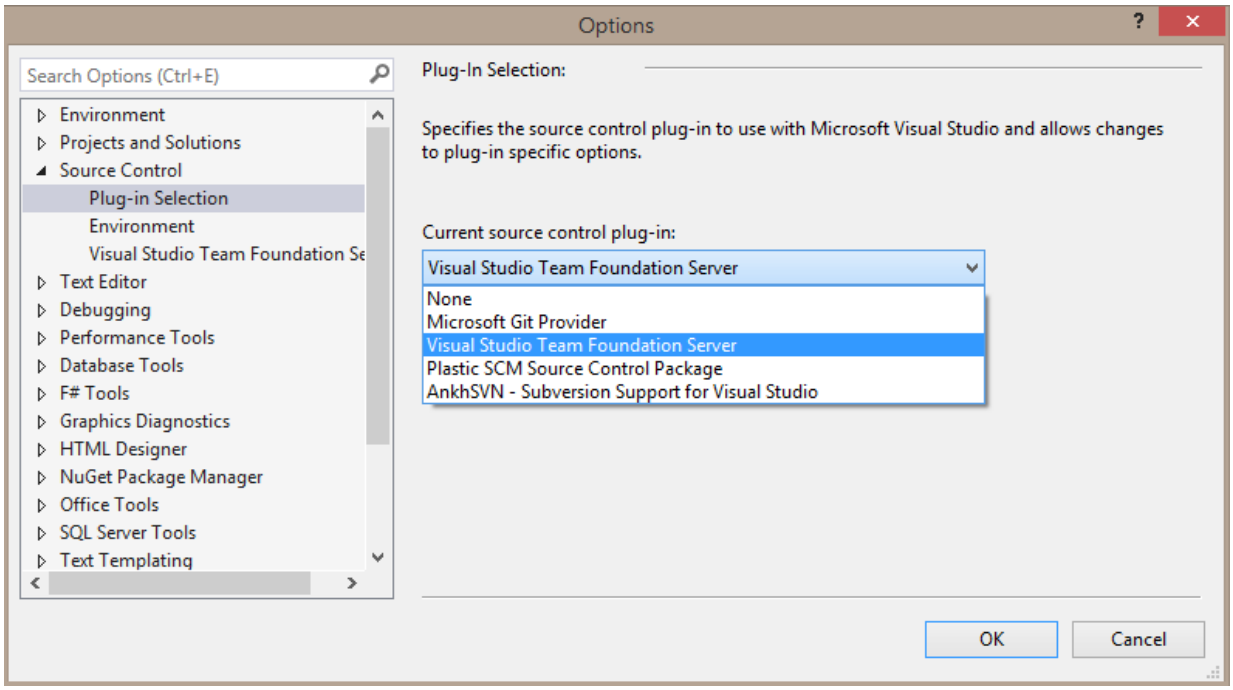
Selection of the Source Control Clients

1. select **Tools** -> **Options** menu in the Microsoft Visual Studio Shell



⇨ The dialogue window shows all the settings for the Microsoft Visual Studio Shell environment.

2. Select "Source Control"



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

3.1 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 of the TcProjectCompare [▶ 18] of the Project Compare tool for further details.

NOTE

Defining of Compare tools

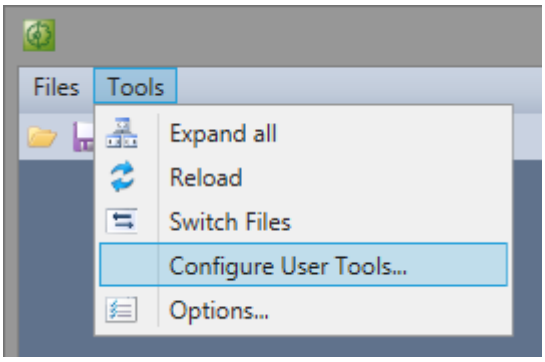
Some clients only allow a general compare/merge tool for all file extensions. For this reason the TwinCAT Project Compare tool offers the option to define further compare tools for special file extensions (see Configuration of the TcProjectCompare for use with source control [▶ 15]).

Configuration of the TcProjectCompare

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

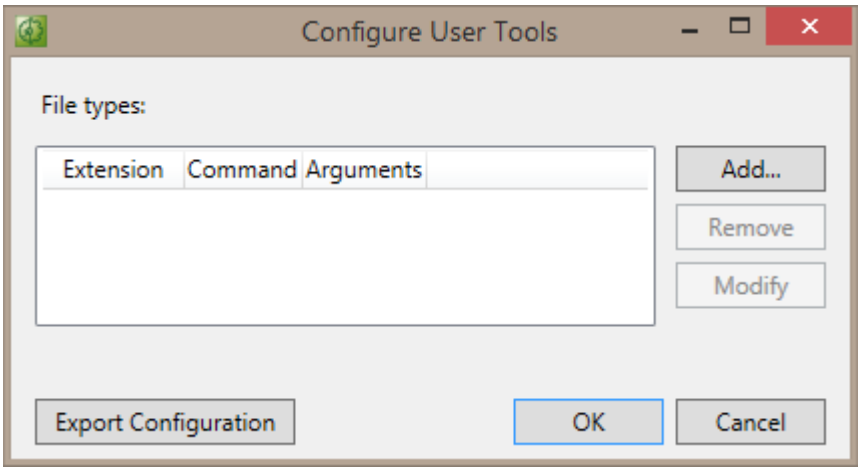
Setting TwinCAT Project Compare as standard tool for "Compare and Merge" functions for subversion AnkhSVN, Microsoft Team Foundation Server or GIT clients:

1. open TwinCAT Compare
2. select the *Tools* -> "*Configure User Tools...*" menu

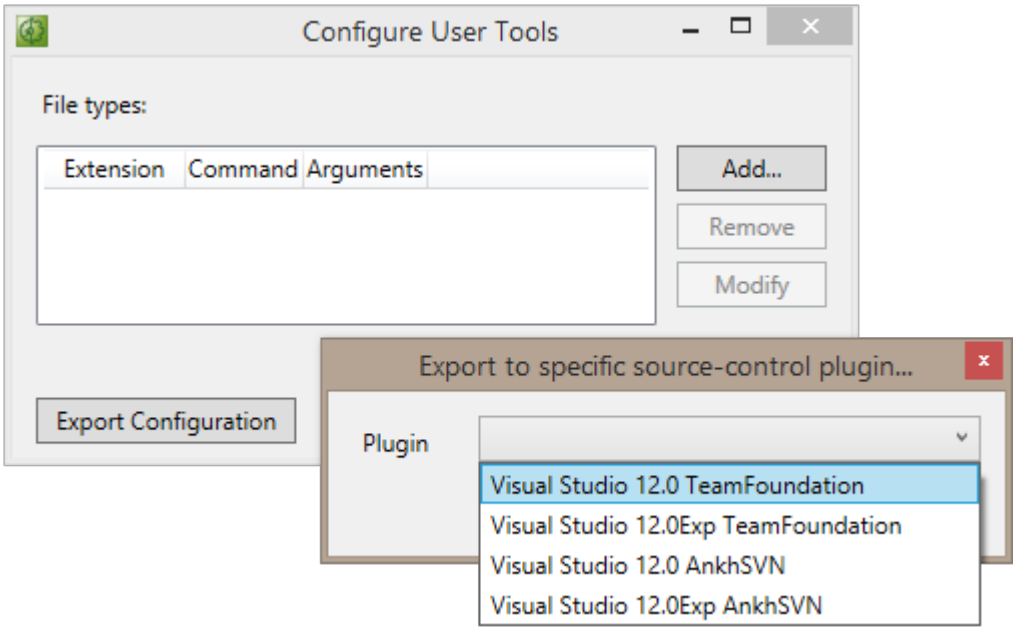


3. in the following dialogue, select a standard tool "Merge". This is used for all areas outside of the PLC (e.g. for the comparison of configurations for tasks).

4. Click on "**Export configuration**"



5. If the required Source Control Plugin has already been selected in Visual Studio Shell, you can export the configuration for this plugin if you are using a Microsoft Team Foundation Server or an AnkhSVN Client.



4 Configuration of the TcProjectCompare for use with source control

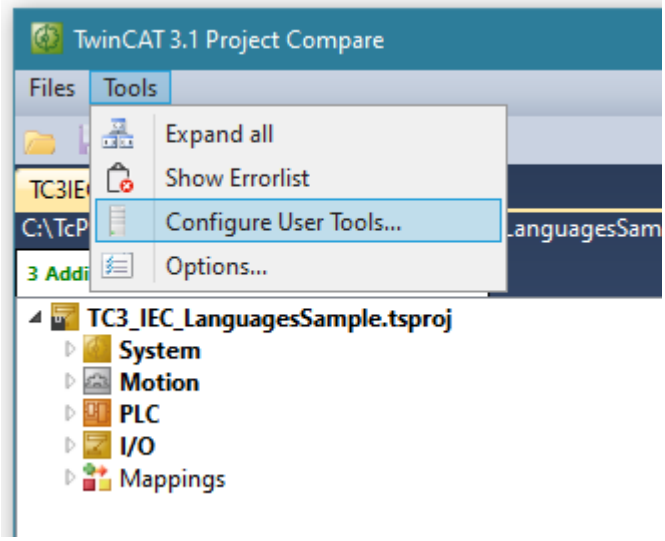
The configuration of the TcProjectCompare can be generated from the software for use with some of the most frequently used source control clients. At the moment, these are the following clients:

- Microsoft Team Foundation Server
- Git
- AnkhSVN

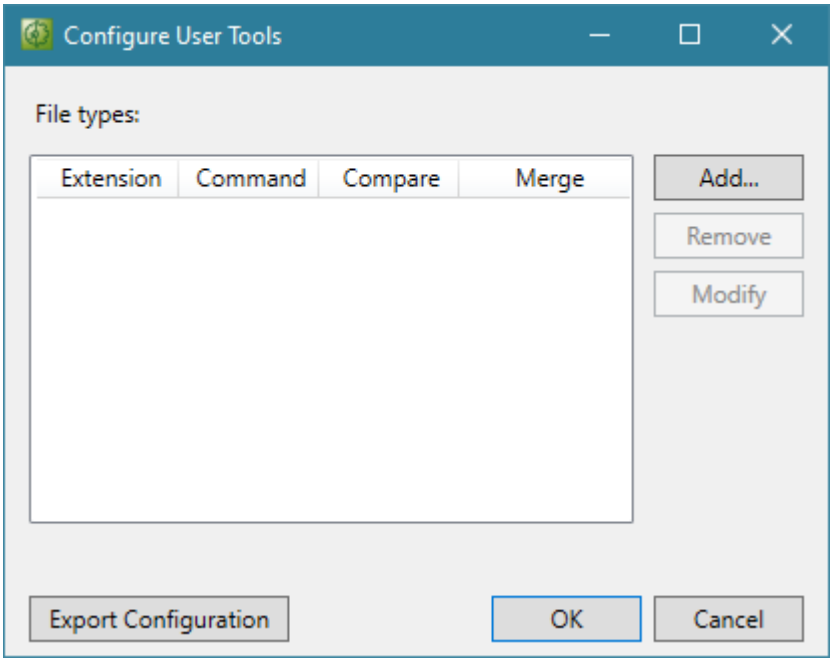
Generating the configuration of these clients

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

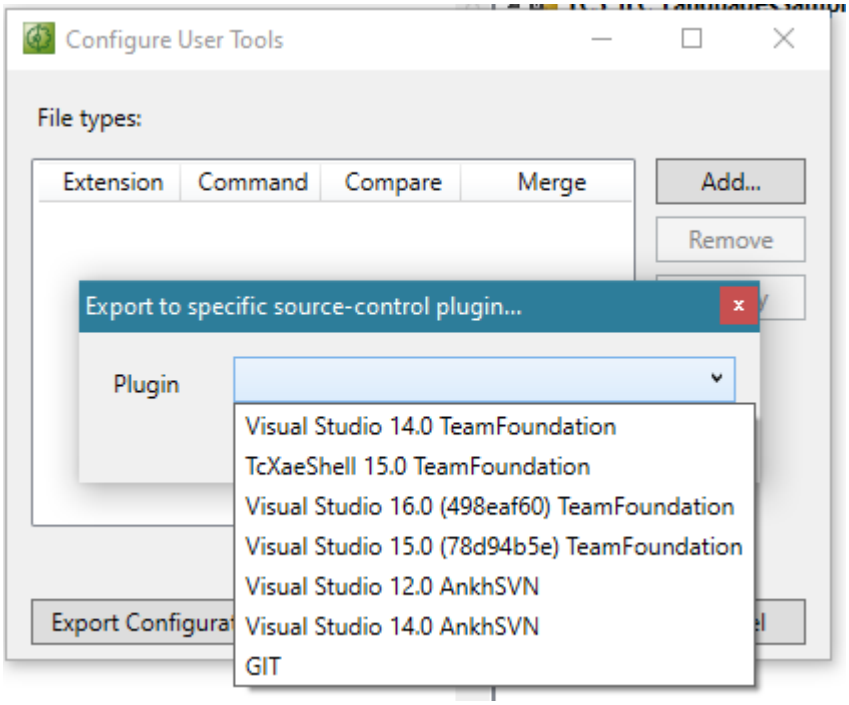
1. Open TcProjectCompare**.**
2. Select the menu **Tools** -> **Configure User Tools....**



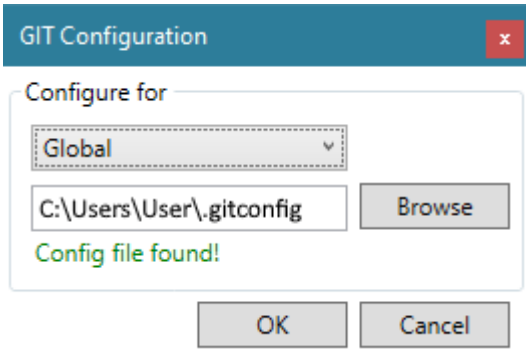
3. Click on **Export Configuration.**



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



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**.



⇨ 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 [▶ 18].

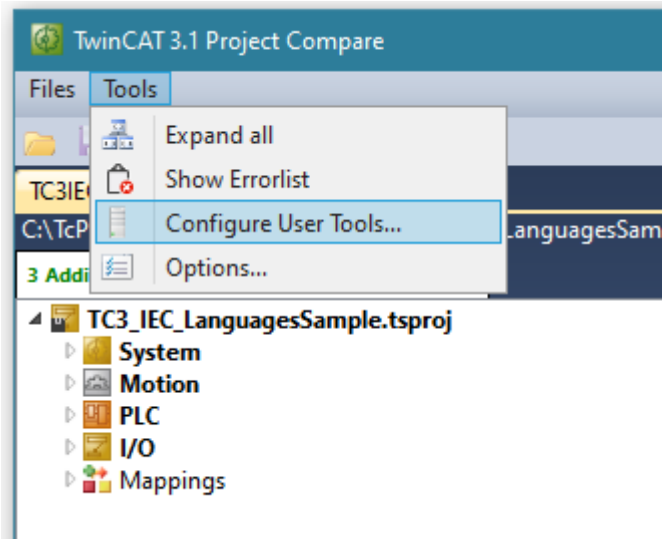
Dealing with non-TwinCAT flies

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.

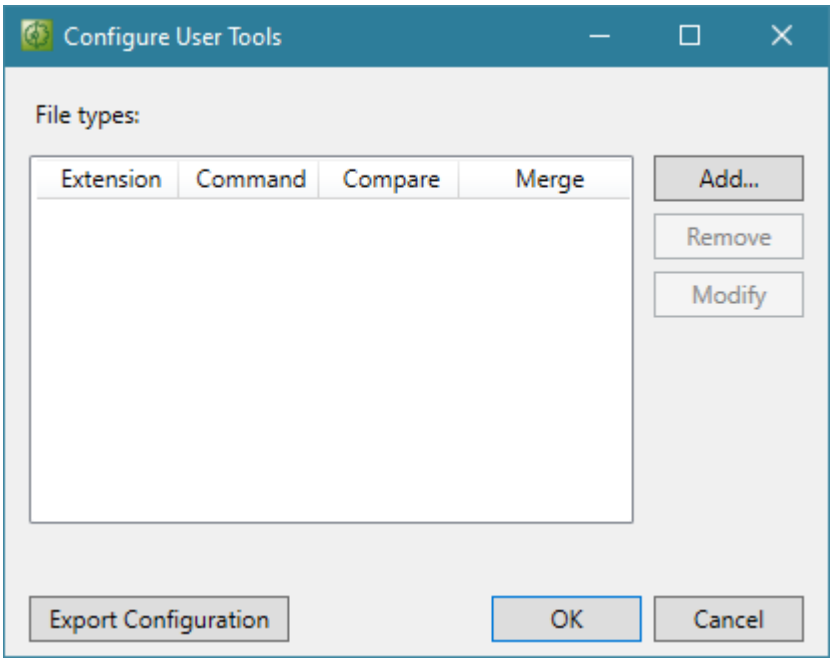
Setting an external compare tool for file extensions:

1. Open the TcProjectCompare.

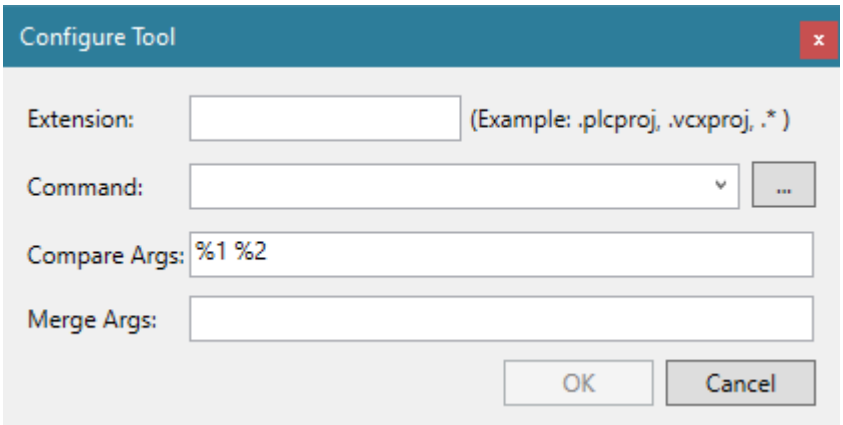
2. Select the menu **Tools** -> **Configure User Tools....**



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**.



4.1 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
```

5 Merge of TwinCAT PLC Projects

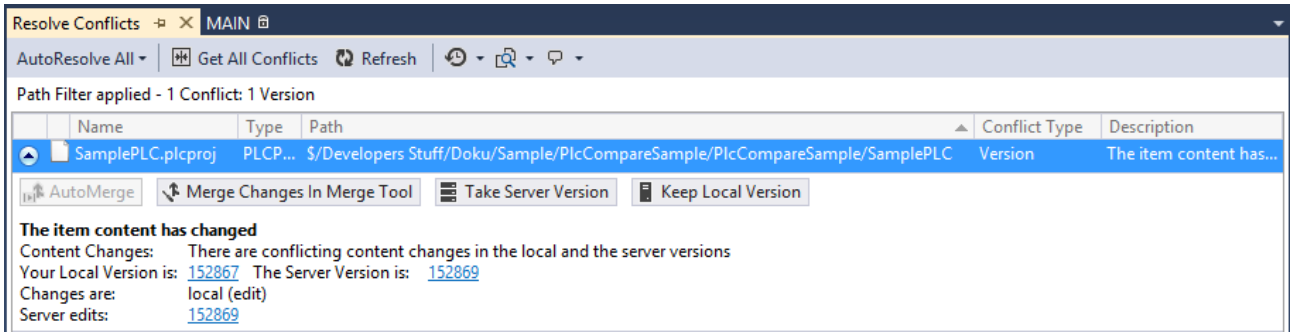
If more than one developer is working on the same PLC project, conflicts may arise when “checking in/ committing” the changes. For example, parallel POUs can be added/edited by more than one developer of the PLC project. These conflicts must be solved if a consistent project is to be created. This procedure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

NOTE

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 must be configured as a merge tool (see Configuration of the TcProjectCompare for use with source control [▶ 15])

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



To solve these problems

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”.

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