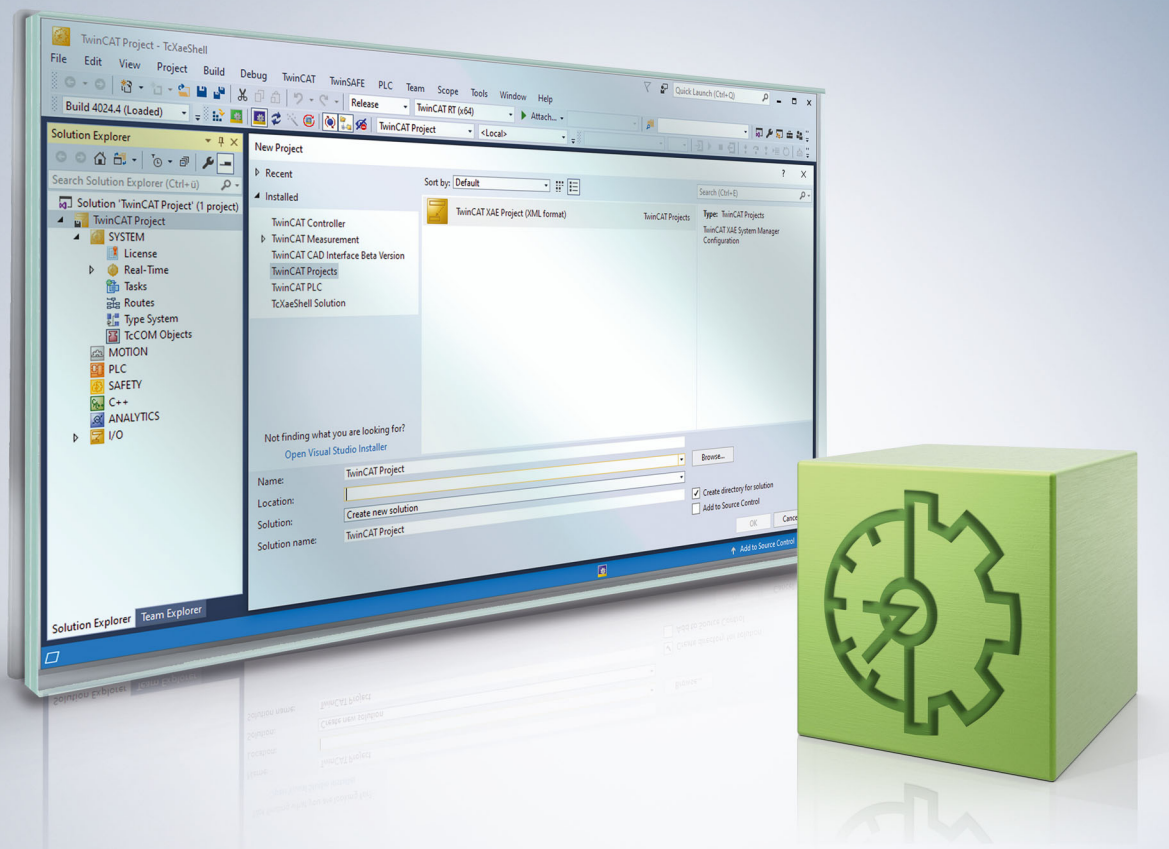
BECKHOFF New Automation Technology

Manual | EN

TE1000

TwinCAT 3 | Multiuser



2025-03-03 | Version: 1.7.1

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.
The documentation and the following notes and explanations must be complied with when installing and commissioning the components.
The trained specialists must always use the current valid documentation.

The trained specialists must ensure that the application and use of the products described is in line with all safety requirements, including all relevant laws, regulations, guidelines, and standards.

Disclaimer

The documentation has been compiled 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.
Claims to modify products that have already been supplied may not 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 the designations or trademarks contained in this publication for their own purposes, this could infringe upon the rights of the owners of the said designations.

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

Copyright

Third-party trademarks

Trademarks of third parties may be used in this documentation. You can find the trademark notices here: https://www.beckhoff.com/trademarks.

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 Installation

As of TwinCAT 3.1 Build 4026 the installation of TwinCAT Multiuser is required. All earlier versions of TwinCAT 3 include TwinCAT Multiuser as part of the standard installation.

TwinCAT Multiuser is installed via the TwinCAT Package Manager. Further information on this can be found in the installation documentation.

You can perform the installation using either the user interface (UI) or the command line interface (CLI).

You can use the UI to display all available workloads and install them as needed. To do this, follow the corresponding instructions in the interface.

You can use the following command to display the available workloads on the system in the CLI:

```
tcpkg list -t workload
```

Explanations on installation via CLI can be found in the chapter Working with the command line.

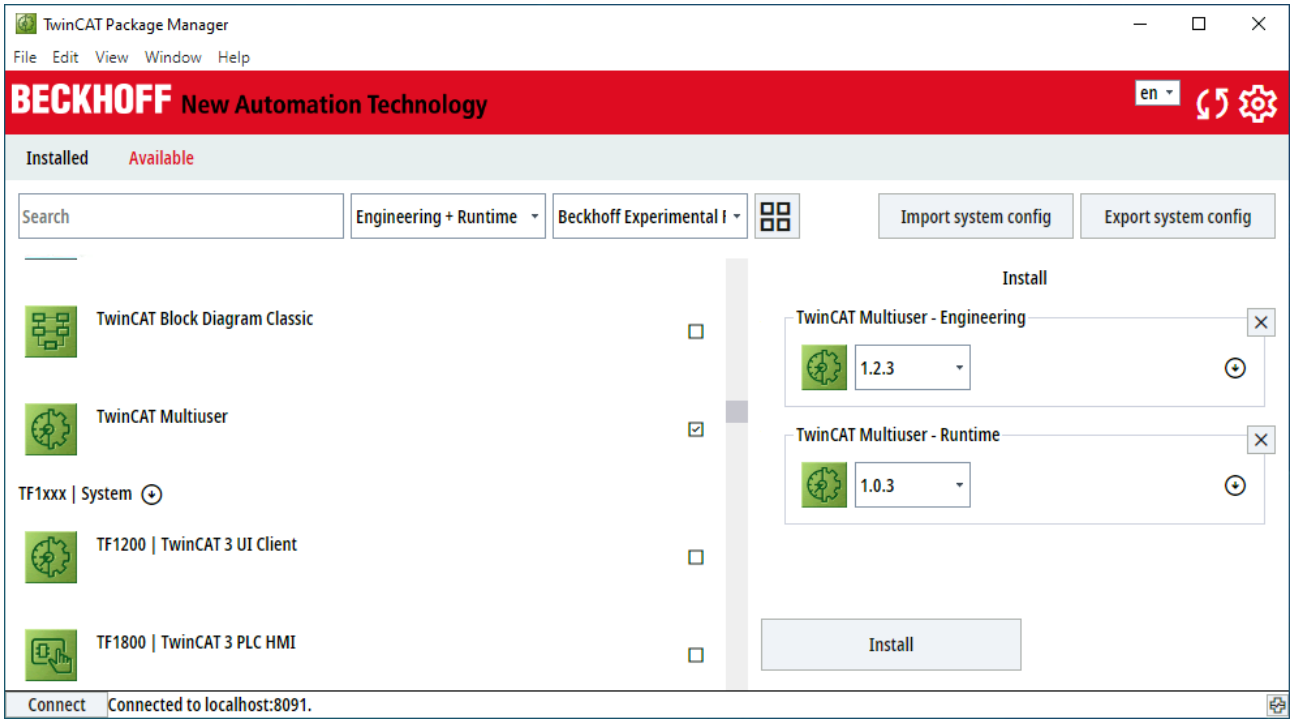
When referring to the TwinCAT Multiuser installation in the following, a distinction is made between PCs with TwinCAT Engineering and systems with TwinCAT Runtime only.

On a PC with Engineering or with Engineering and Runtime

If TwinCAT Engineering and TwinCAT Runtime or only TwinCAT Engineering are installed on your system, install both Multiuser Editor and Multiuser Server.

To do this, choose the following two workloads in the TwinCAT Package Manager UI:

- TwinCAT Multiuser - Engineering
- TwinCAT Multiuser – Runtime



Or install the workloads via the CLI using the following commands:

```
tcpkg install TwinCAT.Multiuser.XAE
tcpkg install TwinCAT.Multiuser.XAR
```

Multiuser Editor and Multiuser Server are installed.

On a runtime PC

Multiuser Server can be used on a runtime-only PC without Visual Studio. In this case, receiving the data and populating the GIT repository is handled by a simplified implementation of the Git commands on the server which maps the minimum requirements for exchange with the engineering system.

Customers who wish to use the full Git functionality in combination with the Multiuser Server are advised to install Git first and then the XAR workload for the Multiuser Server component.

GIT installation:

1. Open the download page on the official GIT website: https://git-scm.com/download/win
2. Depending on your system architecture, choose either the 32-bit or 64-bit version to download.
3. Perform the installation.

> If possible, do not change the installation path, as Multiuser Server expects GIT in a defined path. If the change is still necessary, the corresponding path must be added to the server's gitpath.config file at
> *C:\Program Files (x86)\Beckhoff\TwinCAT\Functions\Multiuser\TcAdsGitServer*

Multiuser installation:

Select the following workload in the TwinCAT Package Manager UI:

- TwinCAT Multiuser – Runtime

Or use the following command to install the workload via CLI:

```
tcpkg install TwinCAT.Multiuser.XAR
```

Multiuser Server is installed.

3 Concept

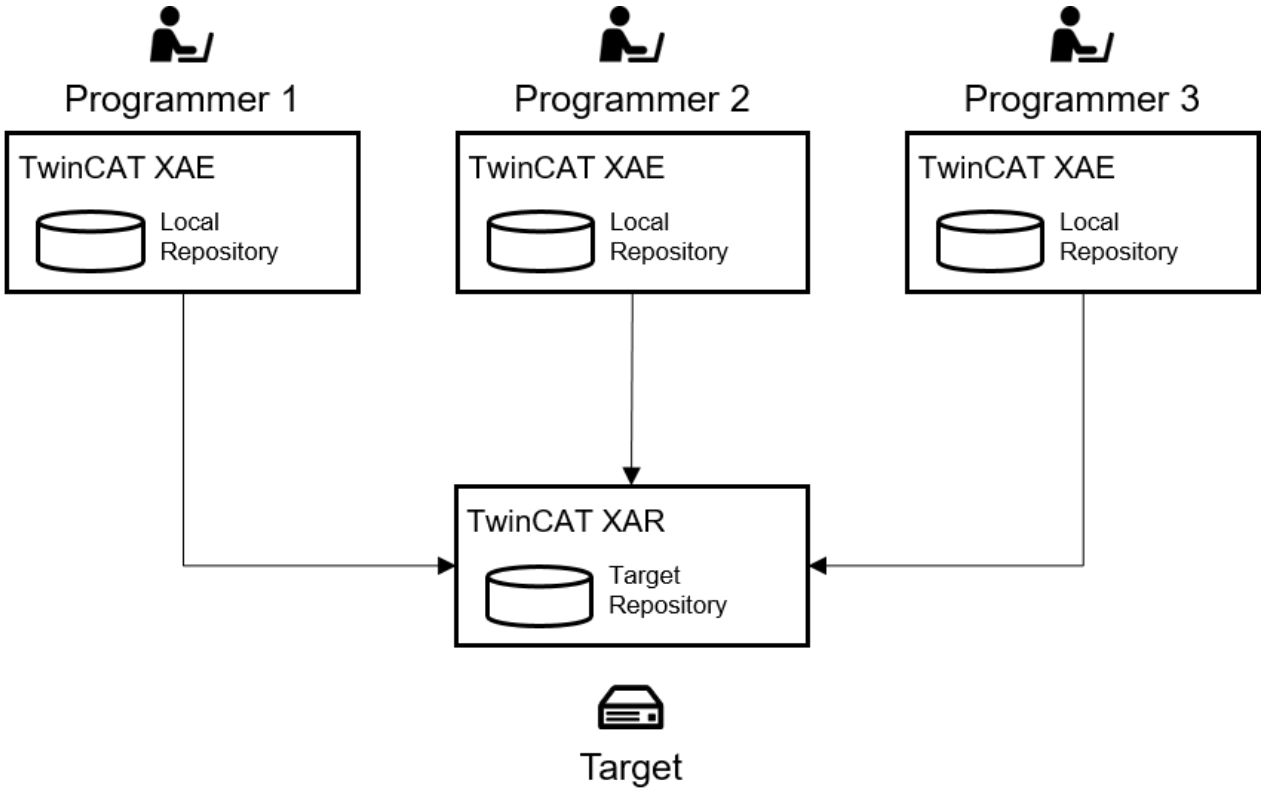
More and more machine and system functions are implemented in software. To allow for this, nowadays teams of several programmers are involved in creating the control code and subsequently the commissioning of these machines and systems. This poses several challenges, especially during commissioning:

- In many cases, access to Source Control Systems is not guaranteed.
- The latest (active) software version on the machine may differ from the local version.
- If changes that are "downloaded" to the target system turn out to be incorrect, it is not possible to undo the "download".

TwinCAT Multiuser was developed to meet these challenges. This is a "local" Source Control System on the target system, the handling of which has been fully integrated into the existing workflow. This means that no in-depth knowledge of the exact functionality of Source Control Systems is required to use TwinCAT Multiuser.

TwinCAT Multiuser is available from version TC3.1.4024.0. In this version, TwinCAT Multiuser mainly facilitates collaboration of several programmers, with each programmer working on one PLC project. If several PLC projects are integrated in a TwinCAT project, the multiuser functionality can be enabled separately for each project. In this case, separate repositories are automatically created for each of these projects, both locally and on the target system.
Since version 3.1.4024.4.0, complete TwinCAT projects can also be managed with the multiuser function. It can now be selected in the settings whether complete TwinCAT projects or only individual PLC projects are managed by the multiuser function.

The following diagram illustrates the concept:



The target system is selected as the Source Control Server during commissioning, since a connection to the target system is required in any case in order to load a control program onto the target system. This avoids the need for additional infrastructure.

Another objective of multiuser integration is to avoid the need for specific Source Control knowledge. The Source Control functionality is integrated in the standard workflow. For example, the current project status is automatically transferred to the target system during a download or online change of a project, without the

need to trigger the transfer separately. The history is also created automatically. All changes are recorded, including user name, timestamp and change. If required, a query for a comment can be set for each change transfer.

To ensure that the history created in this way continues to be available after commissioning, a Source Control System was integrated, which permits several repositories and also transfers the full history into all repositories. After successful commissioning, it is thus possible to transfer the active project status of the machine/system to a "development repository", including traceability of all steps performed during commissioning. The latter is done with the on-board resources of the Source Control System.

The Source Control System on which the multiuser functionality is based is Git.

NOTICE

The TwinCAT 3 Multiuser function uses Git

Git itself is separate and is installed optionally. If a standard Git installation is available on a computer, this can be used for the TwinCAT 3 Multiuser function (see Reference server settings [▶ 29]).

The TwinCAT 3 Installer allows customers who do not have a Git installation on the computer to install a MinGit that is only intended for use from TwinCAT 3 Multiuser and therefore is not entered in the Path environment variable.

Differentiation from the Source Control Integration of TwinCAT:

The general interfacing of TwinCAT with Source Control Systems remains unaffected by the multiuser functionality and can take place independently of it. By using the standard interface of the Microsoft Visual Studio Shell integrated in TwinCAT, a large number of Source Control Systems are available for this purpose. In addition, all aspects of a TwinCAT project can be handled by this integration.

The TwinCAT multiuser functionality is based on the Git standard. As already described, the aim of this functionality is to enable a team of several programmers to collaborate on a control program during the commissioning phase, without the need for a special infrastructure or special knowledge of Source Control Systems on the part of the programmers.

Although it is therefore possible to use different Source Control Systems for both functionalities, the full capabilities can only be achieved if both functionalities are based on Git. This refers to the transfer of the history during commissioning to the "main" Source Control System.

4 Workflow

4.1 Management of a TwinCAT project

4.1.1 Activating the multiuser function for a TwinCAT project

Carry out the following steps to activate this function for a TwinCAT project that is not yet managed through the multiuser functionality.

1. Activate the **Use Multiuser** option in the **Settings** tab of the **SYSTEM** node of the TwinCAT project. (See TwinCAT project settings [▶ 27])
2. Click the **Init** button to make the settings for the multiuser function and initialize the function.
   ⇨ The Settings tab of the **Multiuser Explorer** window opens.
   If you close the window by mistake, you can find it at the menu **TwinCAT -> Multiuser Explorer**.
3. In the **ADS Route** field, select a target system for the multiuser repository.
4. In the **Multiuser Repository** field, select a name for the multiuser repository.
5. If required, select the option **Ask for update message on each usage**. (See also Settings tab [▶ 19])
6. Click the **Init local and remote** button.

⇨ The multiuser repositories in the project and on the target system are initialized.

1. Click the button **Init** once again so that the URL of the multiuser repository is stored in the project.

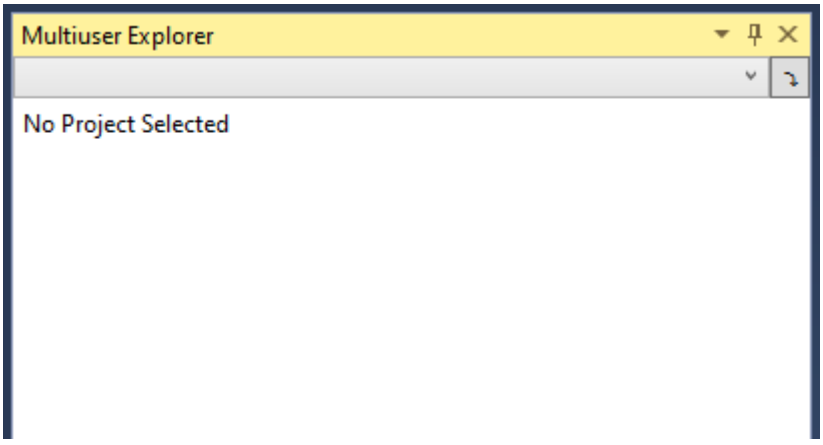
⇨ If the URL is stored directly in the project, this facilitates local initialization for this project on additional systems.

4.1.2 Use of an already set up TwinCAT project on another system

If an existing TwinCAT project for which a repository has already been set up on the target system is to be linked to the multiuser server, proceed as follows:

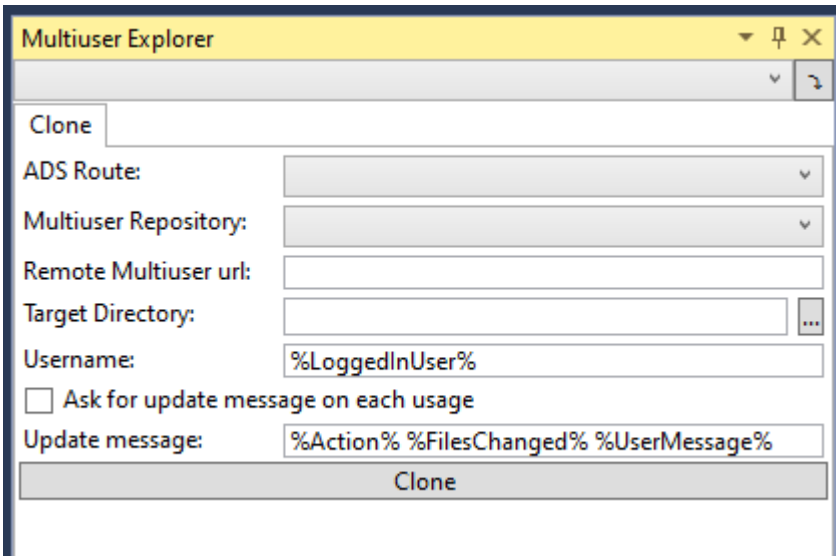
✓ The current status of the project is not available locally.

1. Open the **Multiuser Explorer** from the menu **TwinCAT -> Multiuser Explorer**.
   ⇨ An empty **Multiuser Explorer** opens.



2. Use the **Clone** button  to clone a project from the target system repository to the local system.

⇨ The following view opens.



3. Select the target system from which you want to clone the project.
4. Select the repositoy you want to clone.
5. Select the local folder (Target Directory) to which the project will be cloned.
6. Select the option **Ask for update message on each usage** if you want to be asked for update messages.
7. Click the **Clone** button.

⇨ The project has been copied to the selected folder and can be opened for further editing.

⇨ The multiuser function is already switched on and configured.

✓ The current status of the project is already available via a Source Control system.

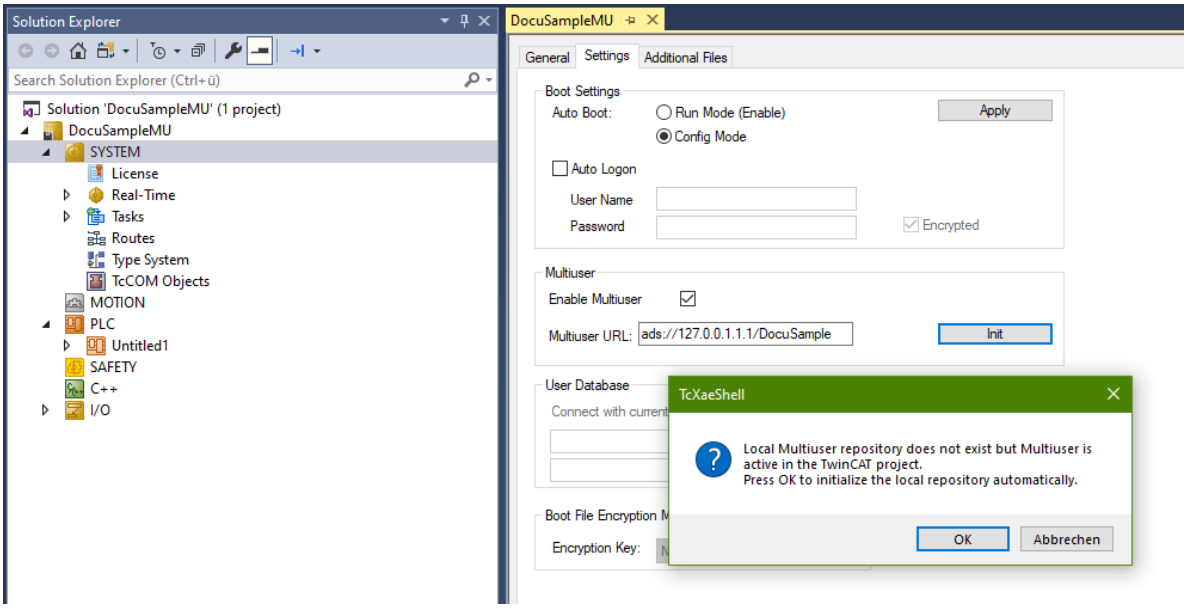
1. Open the TwinCAT project.
2. In the **Settings** tab of the project's SYSTEM node, check whether the **Use Multiuser** option (see TwinCAT project settings [▶ 27]) is enabled and the multiuser URL is stored in the project.

⇨ Depending on whether the multiuser URL is stored in the TwinCAT project, the further procedure differs.

✓ The multiuser URL is stored in the project.

1. Click the **Init** button to initialize the multiuser function locally.

⇨ The following dialog opens.



2. Click the **OK** button.

⇨ The multiuser function is initialized locally for the TwinCAT project and can be used.

✓ The multiuser URL is not stored in the project.

1. Click the **Init** button.
   ⇨ The **Settings** tab of the **Multiuser Explorer** opens automatically.
2. In the **ADS Route** field, select the target system that contains the multiuser repository.
3. In the **Multiuser Repository** combo box, select the name of the multiuser repository to which the project is to be linked.
4. If required, select the option **Ask for update message on each usage**. (See also Settings tab [▶ 19].)
5. Click the **Init local and remote** button.
   ⇨ The multiuser repository in the TwinCAT project is now initialized.

4.2 Management of a TwinCAT 3 PLC project

4.2.1 Activating the multiuser function for a PLC project

To activate this function for a PLC project that is not yet managed through the multiuser functionality, carry out the following steps:

1. Activate the **Use Multiuser** option in the project settings of the PLC project that is to be managed through the multiuser functionality. (See PLC project settings [▶ 27])
2. Click the **Init** button to make the settings for the multiuser function and initialize the function.
   ⇨ The **Settings** tab of the **Multiuser Explorer** window opens automatically.
   If you close the window by mistake, you can find it under the menu **TwinCAT -> Multiuser Explorer**.
3. In the **ADS Route** field, select a target system for the multiuser repository.
4. In the **Multiuser Repository** field, select a name for the multiuser repository.
5. If required, select the option **Ask for update message on each usage**. (See also Settings tab [▶ 19])
6. Click the **Init local and remote** button.

⇨ The multiuser repositories in the project and on the target system are initialized.

1. Click the button **Init** once again so that the URL of the multiuser repository is stored in the project.

⇨ If the URL is stored directly in the project, this facilitates local initialization for this project on additional systems.

Also see about this

- Reference project settings [▶ 27]

4.2.2 Use of an already set up PLC project on another system

If an existing PLC project is to be linked to the multiuser server for which a repository has already been set up on the target system, proceed as follows:

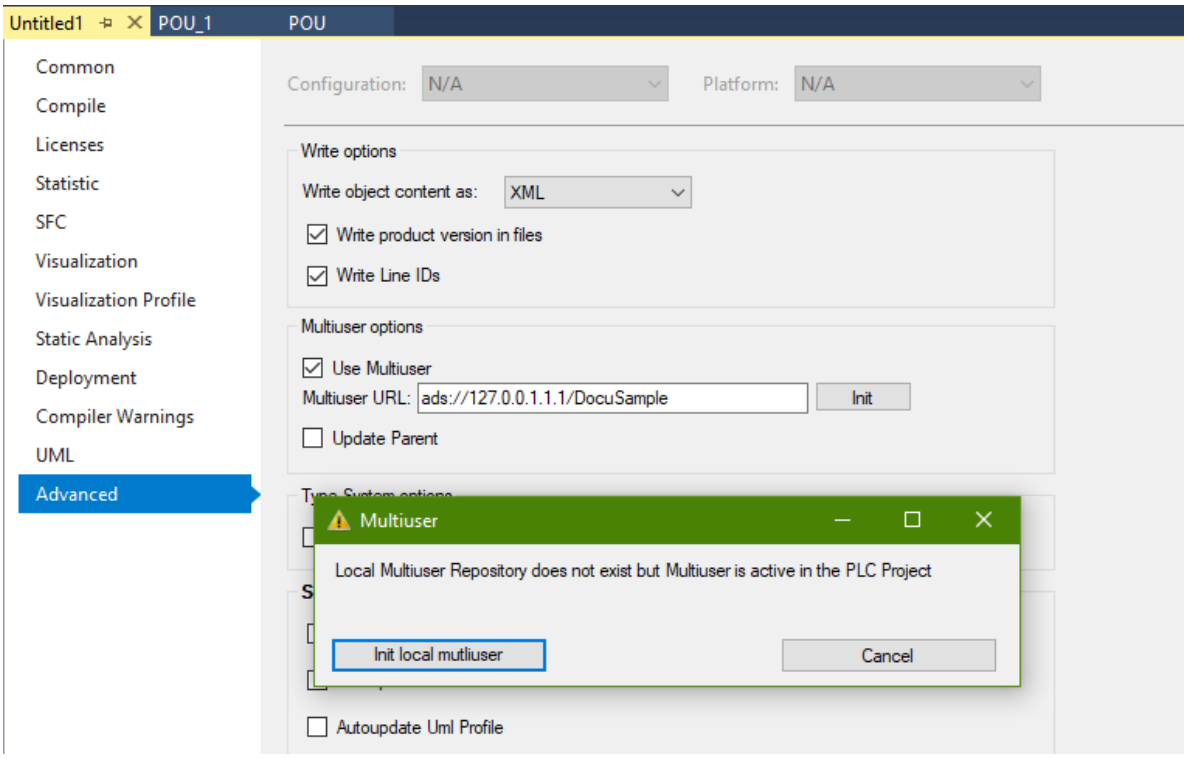
1. Open the project.
2. In the project settings of the PLC project that is to be managed by the multiuser functionality, check whether the option **Use Multiuser** (see Reference project settings [▶ 27]) is activated and the multiuser URL is stored in the project.

⇨ Depending on whether the multiuser URL is stored in the TwinCAT project, the further procedure differs.

✓ The multiuser URL is stored in the project.

1. Click the **Init** button to initialize the multiuser function locally.

⇨ The following dialog opens.



2. Click the **Init local multiuser** button.

⇨ The multiuser function is now initialized locally for the PLC project and can be used.

✓ The multiuser URL is not stored in the project:

1. Click the **Init** button.
   ⇨ The **Settings** tab of the **Multiuser Explorer** opens automatically.
2. In the **ADS Route** field, select the target system that contains the multiuser repository.
3. In the **Multiuser Repository** combo box, select the name of the multiuser repository to which the project is to be linked.
4. If required, select the option **Ask for update message on each usage**. (See also Settings tab [▶ 19])
5. Click the **Init local and remote** button.
   ⇨ The multiuser repository in the project is now initialized.

If you do not have an existing project, you can create a TwinCAT project. Add an empty PLC project and link it to the multiuser repository on the target system as described above.

Git uses absolute paths in the repository

The Git Source Control System uses absolute paths in the repositories. This means that copying a project in which multiuser functionality is already installed (including the .TcGit folder) will result in the multiuser functionality not working correctly.

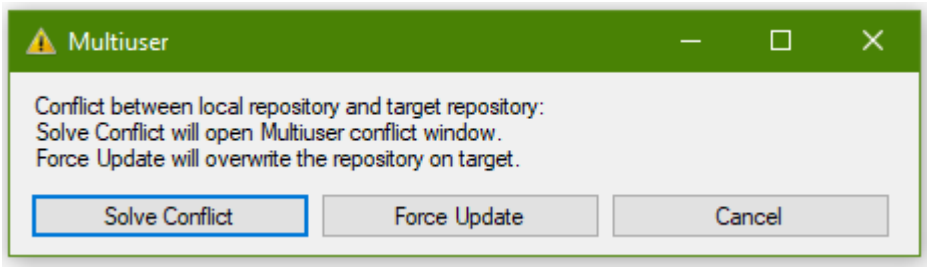
4.3 Working with set functionality

If the multiuser functionality has been set up as described in the previous chapters, this functionality integrates automatically into the existing workflow.

Transferring the data to the target repository

The changes generated while working on the project are automatically transferred to the target repository during an Activate, Download or Online Change if multiuser functionality is activated.

Checks are carried out to ascertain whether there is a conflict with the data in the target repository. This is the case if other programmers working on the project have already made changes to the same project parts (e.g. POUs, settings etc.) and transferred them to the target repository. If there is a conflict, it is displayed in the following dialog.



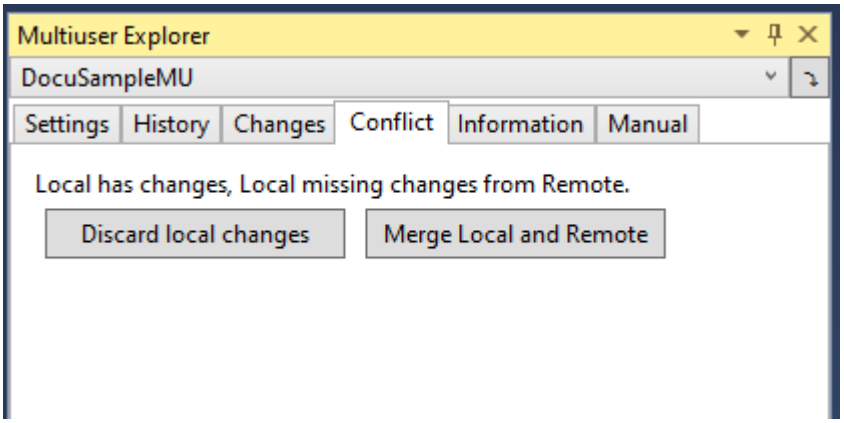
In this selection dialog you then have the option to

- resolve the conflicts using the TwinCAT Project Compare Tool,
- force an update of the target system including the target repository, or
- cancel the Activate, Download or Login.

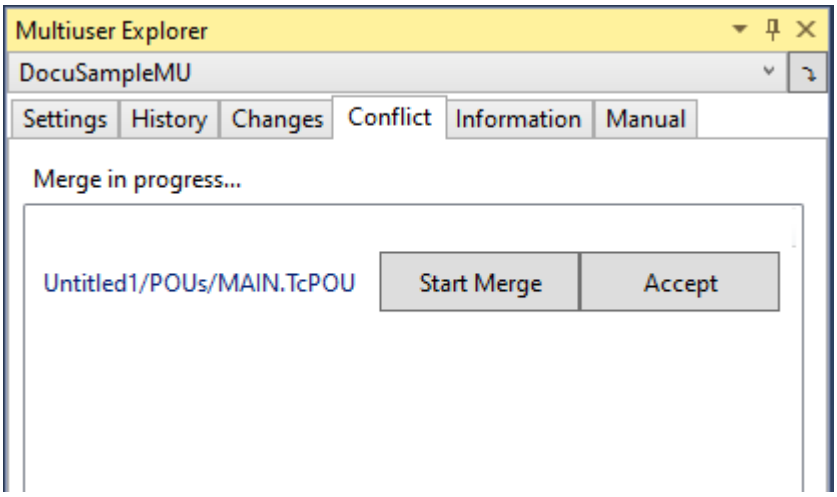
The Force Update button can be used if the target system is required to run with the current project without taking into account the changes made by the other programmers involved in the project. There is no synchronisation with the multiuser repository. Once the project is running, the synchronisation with the multiuser repository must be done manually at a later time (see Manual tab [▶ 25]).

The normal workflow for collaborating on a project involves resolving conflicts. In this step, you can merge the changes you have made to the project with those of the other programmers working on the project.

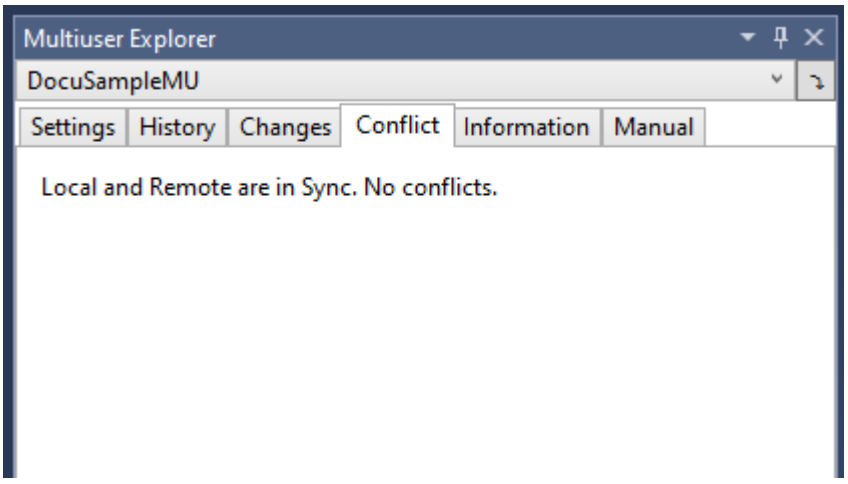
1. To do this, click **Solve Conflict**.
   ⇨ The **Conflict** tab now opens.
   This allows you to discard all your local changes and apply the version from the target repository, or to merge the two versions.
2. To merge the versions, click **Merge Local and Remote**.



⇨ The subsequent overview shows you all files (e.g. POUs (Programming Organization Unit)) where changes have led to conflicts with the target repository.



3. For each conflict displayed, click **Start Merge**.
   ⇨ The TwinCAT Project Compare Tool opens, in which you can merge the changes.
4. Accept the result with **Accept**.
   ⇨ If the merge is confirmed in the TwinCAT Project Compare Tool and the tool is closed, the accept of the merge takes place automatically from TwinCAT version 3.1.4024.40 onwards. If you want to accept the changes directly, confirm it with the **Accept** button. Proceed in the same way in the previous TwinCAT versions.
   ⇨ Once all changes have been merged, you are notified that there are no further conflicts.



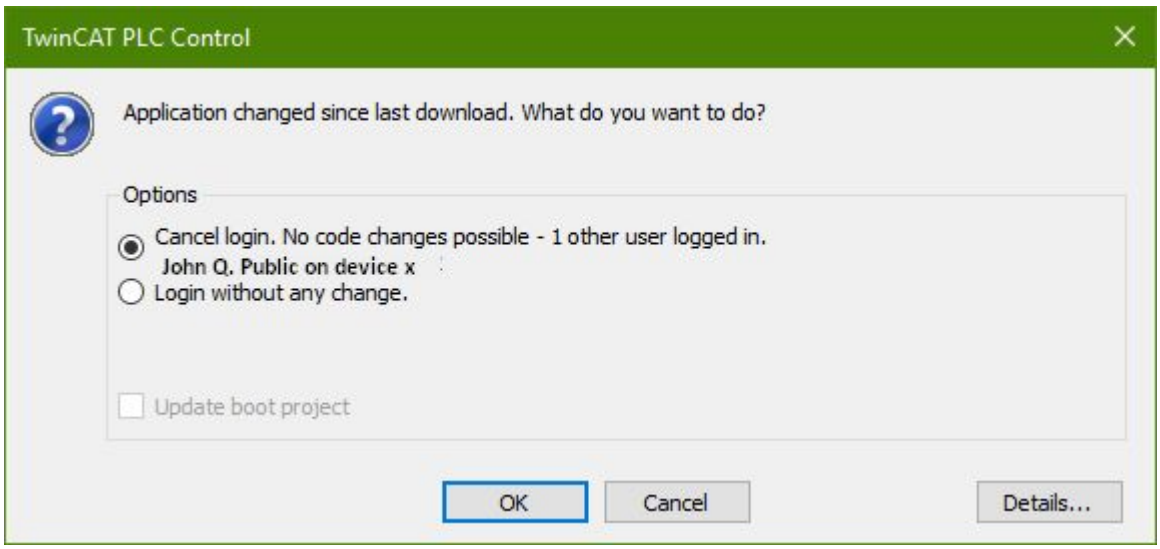
5. Now perform an Activate, Download or Login again.

⇨ The current and merged version of the project is transferred to the target system and stored in the target repository.

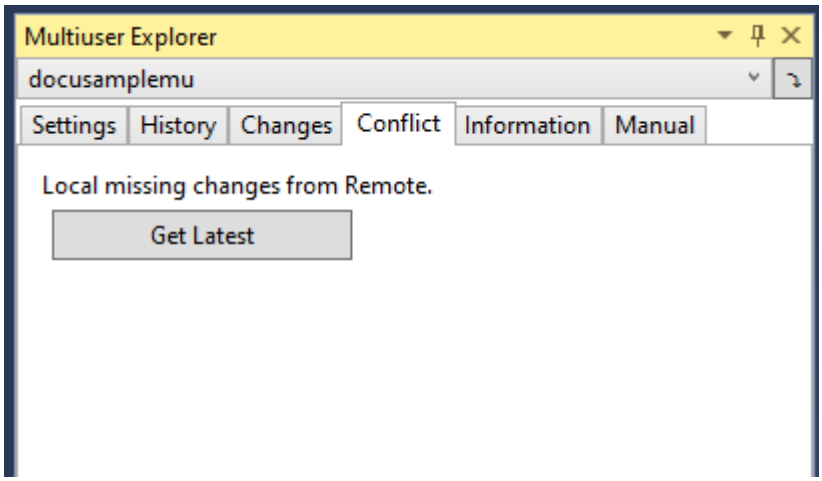
No simultaneous changes by multiple programmers

Simultaneous changes by several programmers are not possible.

Although several programmers can be logged into the target system at the same time to view the project status, only one programmer can make changes at any one time. All other programmers have to log out. The logged-in programmers are shown in the following dialog.

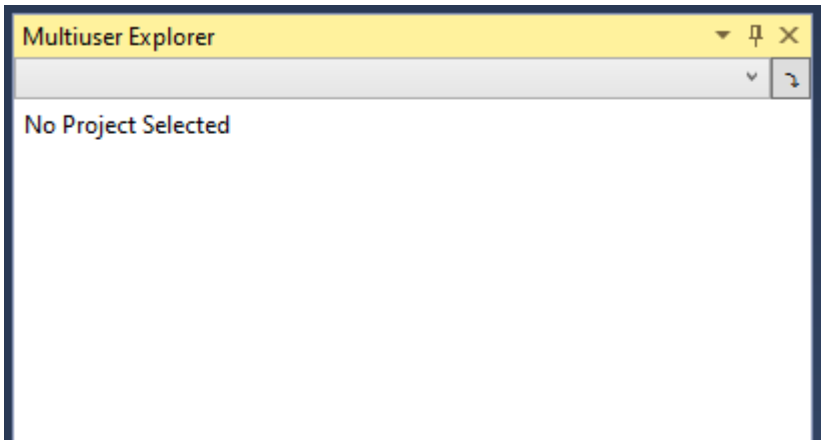


Once the changes have been made, the other programmers can log in again and the workflow shown above is triggered to merge the changes with those of the other programmers. If the other programmers have not yet made any further changes to the project, they are notified that a newer version is available on the remote system.



5 Reference user interface

The Multiuser Explorer is the central administration tool for the multiuser functionality. This window remains empty as long as no project has been associated with multiuser functionality.



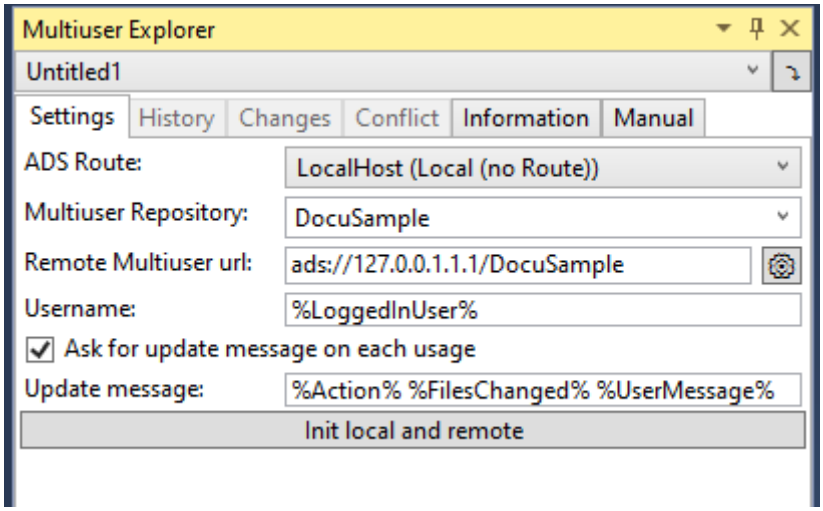
	Using the **Clone** button, the status of the project can be retrieved from the target system repository on an engineering system and stored locally.

Once a project is selected, the tabs described in this chapter become visible.

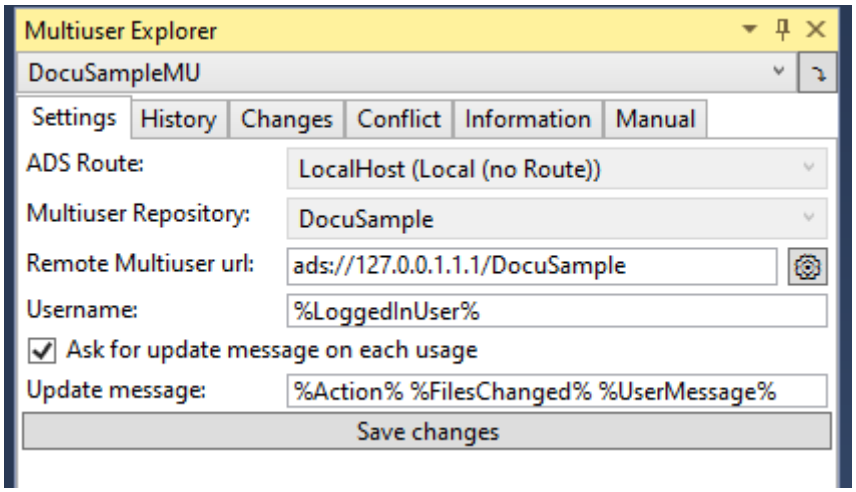
5.1 Settings tab

Once a project has been selected, the **Settings** tab appears as follows:

- If the multiuser function is not yet initialized:



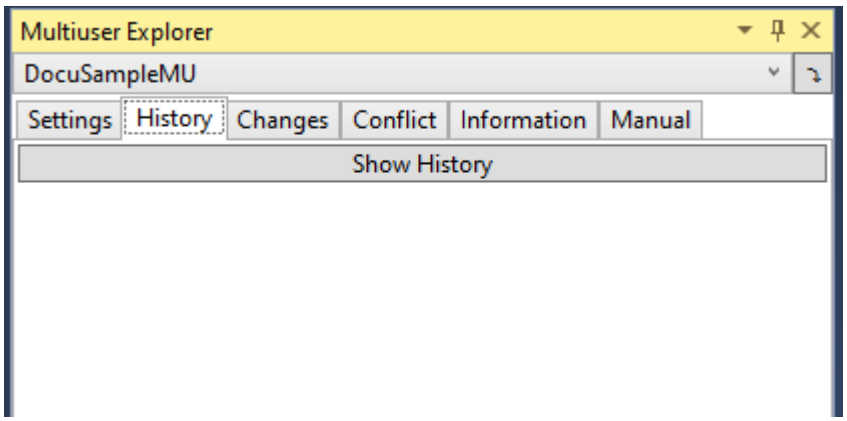
- If the multiuser function is already initialized:



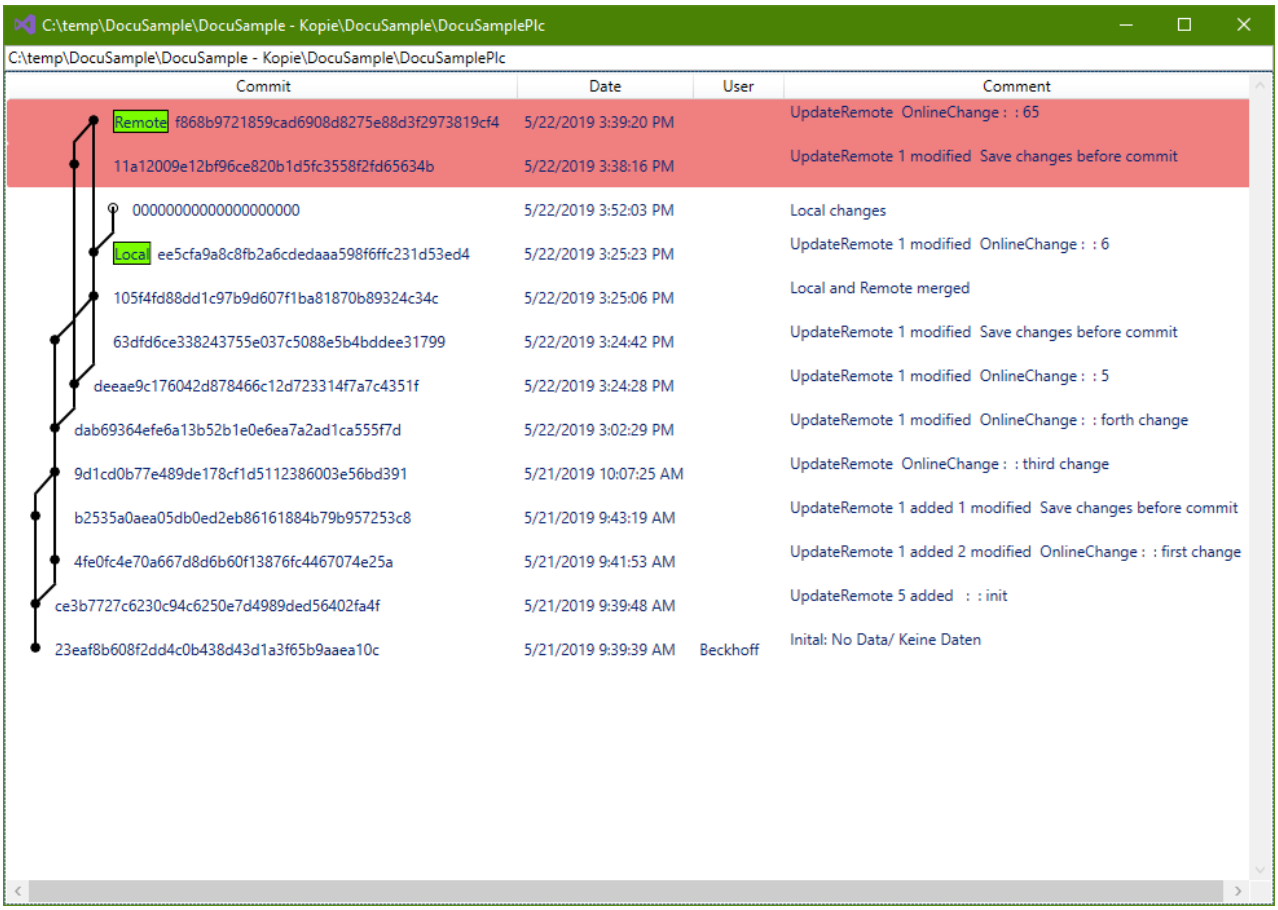
ADS Route	Combo box for selecting an existing ADS route
Multiuser repository	Combo box for selecting an existing multiuser repository or creating a new one.
Remote Multiuser URL	URL for repository
User name	User name to be used for the entries in the history. The default setting is the placeholder "%LoggedInUser%“. This is automatically replaced by the locally logged in user as user name in the comment.
Ask for update message on each usage	Option to query change messages.
Update message	Structure of the update message The placeholders "%Action%", "%FilesChanged%" and "%UserMessage%" are automatically replaced by the corresponding information during an online change or download. If a placeholder is removed, the corresponding information is removed from the automatically generated comment. Placeholder: "%Action%": indicates whether an online change or download was performed. "%FilesChanges%" indicates the number and reason for the changed files (e.g. 2 added). "%UserMessage%" is replaced by the user-specific comment if the option **Ask for update message on each usage** is activated.
Init local and remote	Initializes the multiuser function with the selected settings.
Save changes	Saves the changes made to the settings.

5.2 History tab

The **History** tab has a **Show History** button. This opens a tool window which shows the history of the current project.



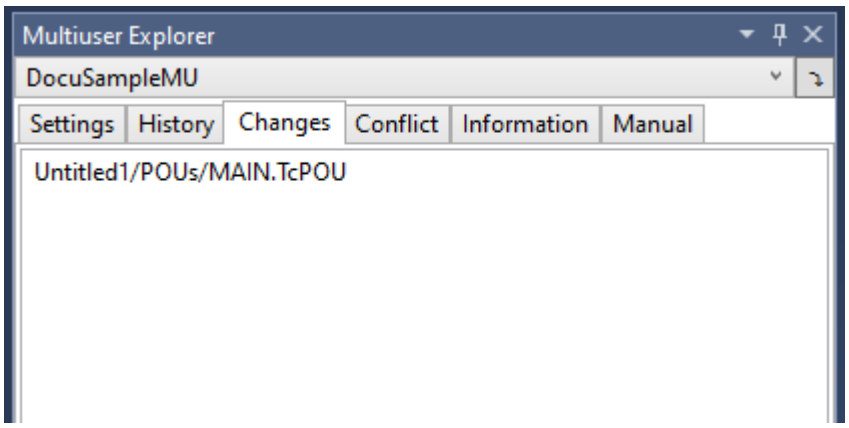
Example: History of the documentation example. In this example, changes were made both on the target system and on the local system.



5.3 Changes tab

The **Changes** tab shows all files where changes have been made that have not yet been transferred to the target repository.

In the following image, exemplary changes have been made to the MAIN-POU (Program Organization Unit) of the Untiteld1 project.



Updating of the target system only in case of changes in the code

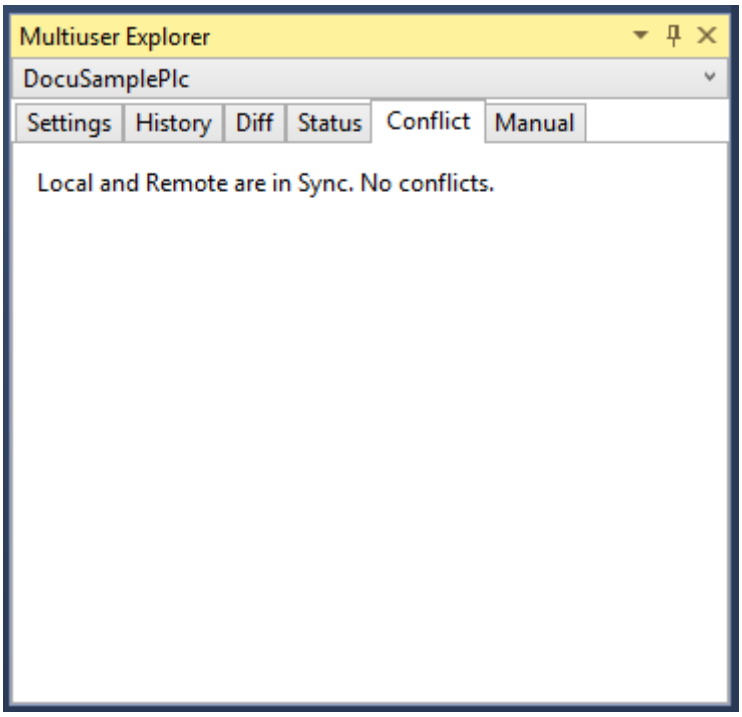
Not every change displayed is a code change. The changes are only transferred to the target system if there is a change in the code that requires a new download of the project or project parts.

Example: A change in a comment within a POU is a change within a file that does not require recompiling. For this reason, it is possible to log in without online change and thus without downloading the changes.

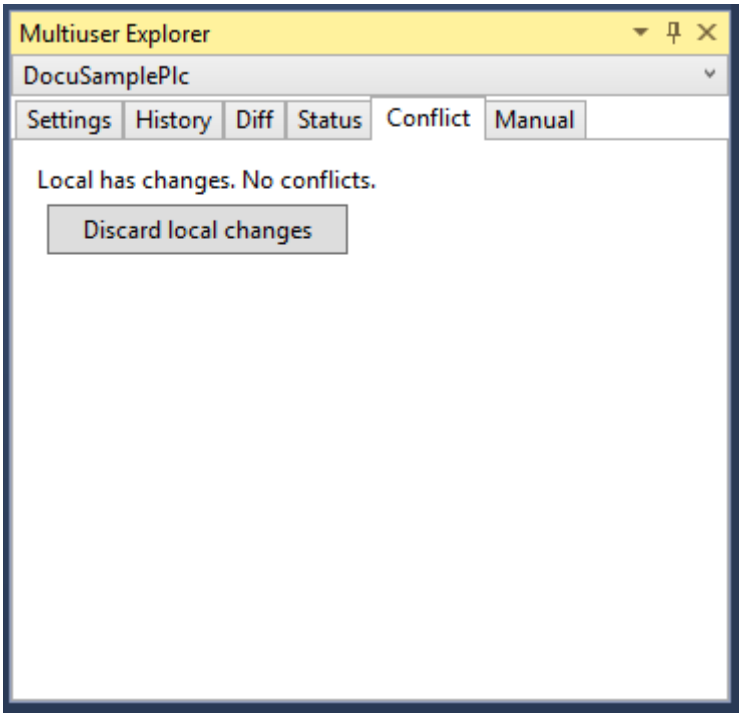
5.4 Conflict tab

The **Conflict** tab shows whether there are any conflicts between the local status and the status in the target repository, or if a merge has been triggered. The following cases are possible:

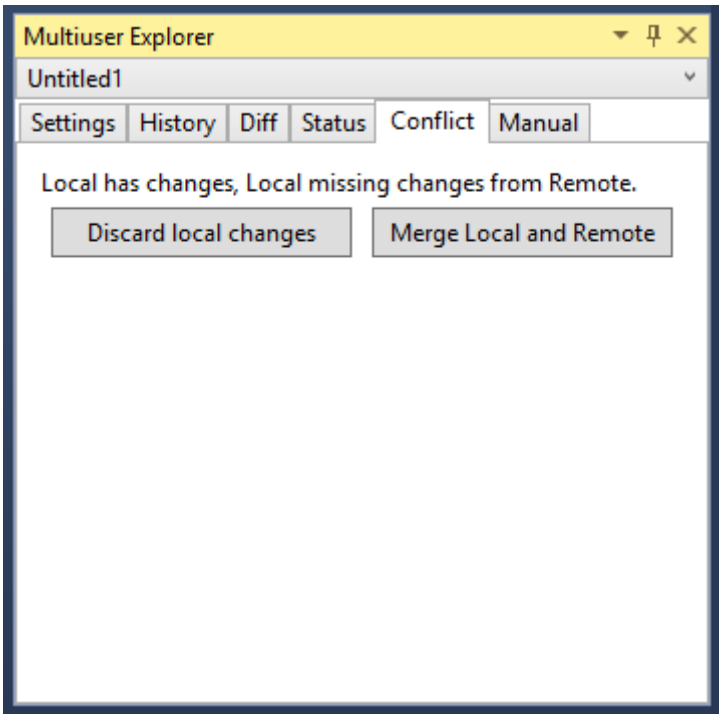
Local and target repositories have the same status:



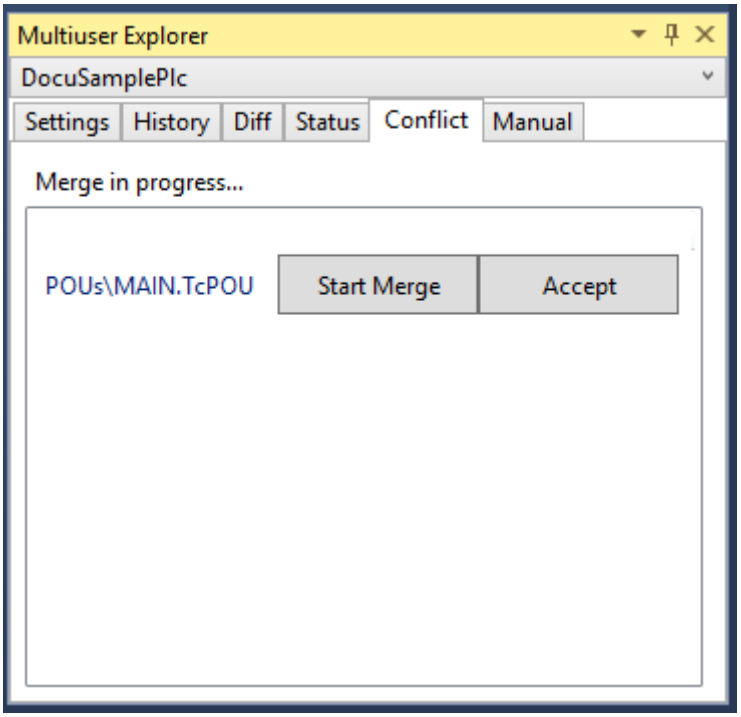
The local system contains changes that have not yet been transferred:



Changes were made on both the local and the remote system:

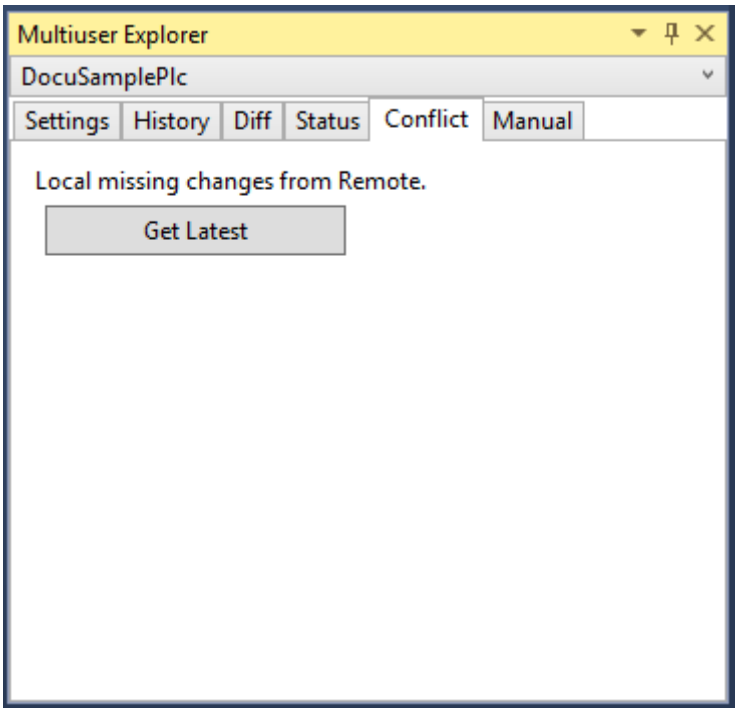


Click the **Merge Local and Remote** button to show a list of the changes. In the following view you will see all objects in which changes have been made on both sides (for example the POU Main in the following figure).



To merge the changes, click the **Start Merge** button. The TwinCAT Project Compare Tool opens, in which you can merge the objects. After a successful merge, please confirm the changes with **Accept**, both in the TwinCAT Project Compare tool and in the Multiuser Explorer.

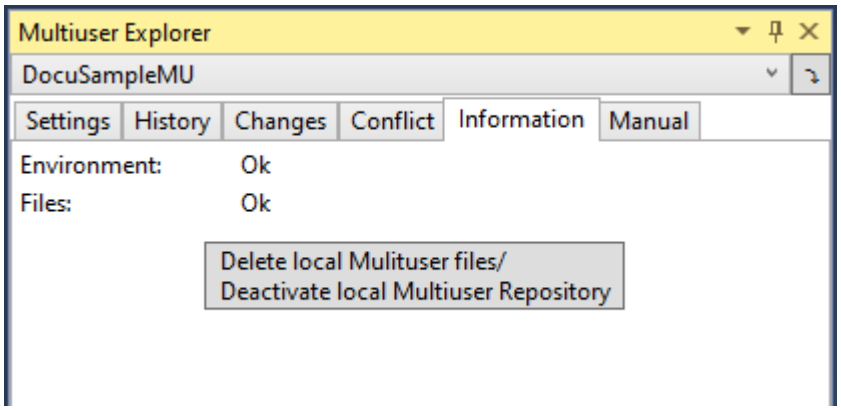
The target repository contains changes that have not yet been implemented locally:



Click the **Get Latest** button to fetch the current status of the target repository. TwinCAT now informs you that files have changed in the background and asks whether you want to reload them. Confirm this question with **OK**.

5.5 Information tab

Displays the status of the multiuser functionality and provides the option to disable this setting. This will delete the local repository.

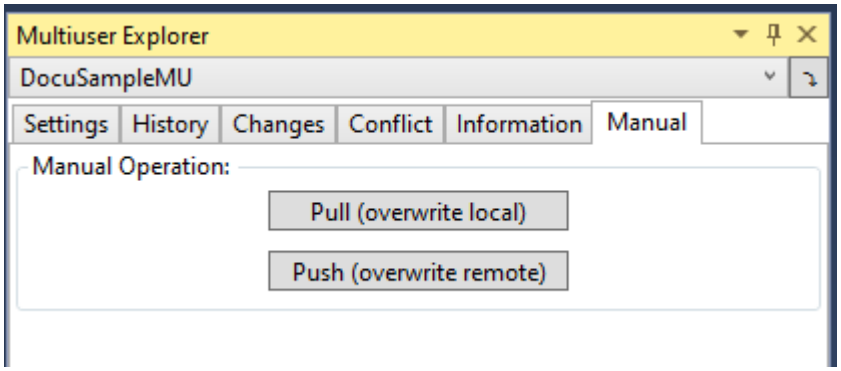


Environment	Displays the state of the multiuser server: **OK**: local and target repository found. **LocalNotExists**: no local repository was found. **RemoteNotExists**: no target repository was found.
Files	Shows the status of the files: **OK**: local and target repository found, no newer project status on target repository. **LocalNotExists**: no local repository was found. **RemoteNotExists**: no target repository was found. **LokalsbehindRemote**: a newer version exists on the target repository.
Delete Multiuser files / Deactivate Multiuser	Deletes the local repository and deactivates multiuser functionality.

The **Files** status entry also displays **OK** if the local files status is newer than that in the repository. You can see which files have been changed in the Diff [▶ 21] tab.

5.6 Manual tab

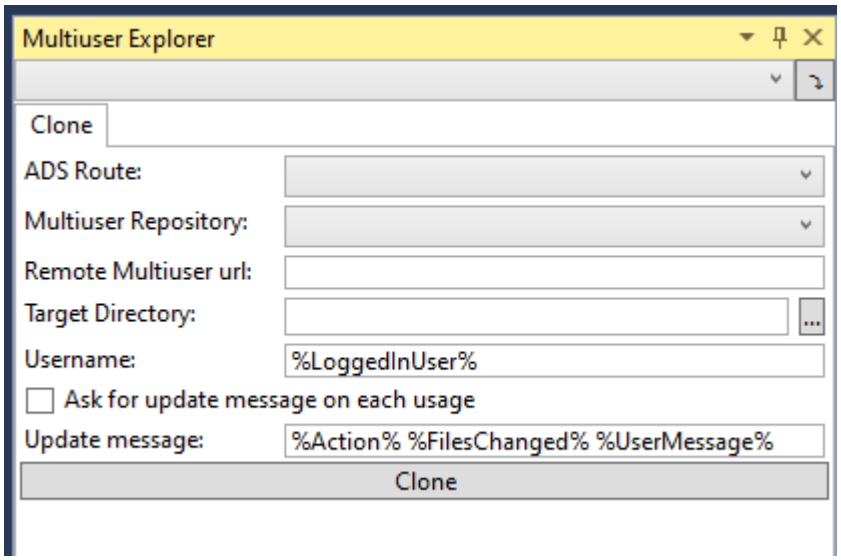
The **Manual** tab is used to write the current project status to the target system or to fetch it from there.



Pull (override local)	Fetches the current project status from the target system and overwrites the local project. There is no merge of the project statuses.
Push (override remote)	Overwrites the status on the target system. There is no merge of the project statuses.

5.7 Project cloning from target system

If you have selected in the Multiuser Explorer that an existing project is to be cloned from the target system (using the button ), then the following view opens in the Multiuser Explorer:

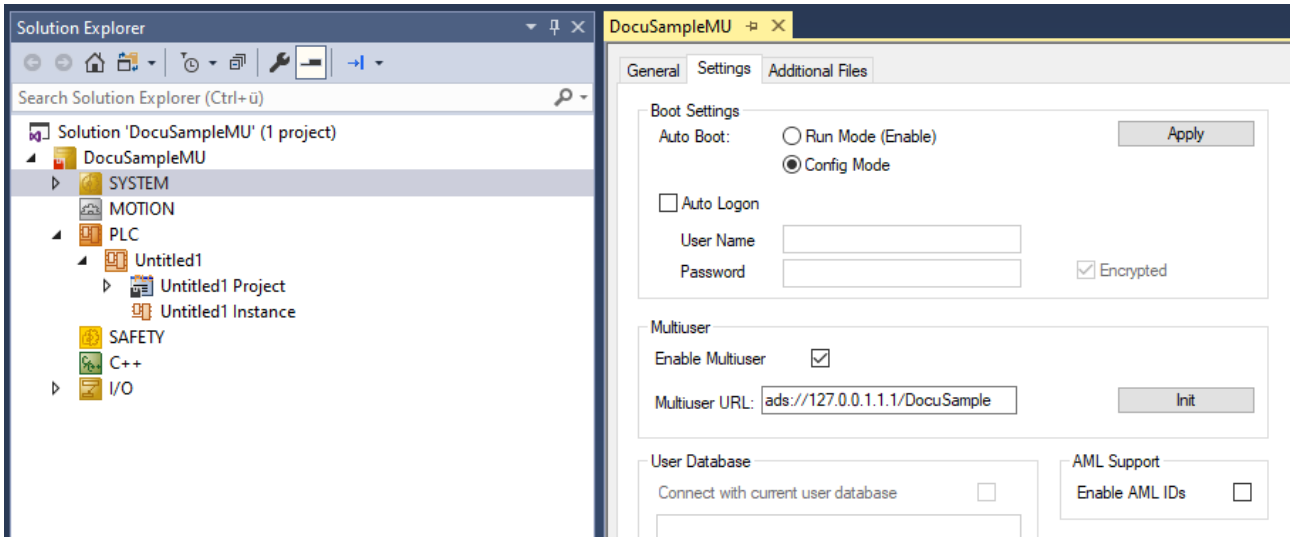


ADS Route	Combo box for selecting an existing ADS route
Multiuser repository	Combo box for selecting an existing MU repository
Remote Multiuser URL	URL for repository
Target Directory	Selection of the target directory to which the project is to be cloned
User name	User name to be used for the entries in the history. The default setting is the placeholder "%LoggedInUser%“. This is automatically replaced by the locally logged in user as user name in the comment.
Ask for update message on each usage	Option to query change messages.
Update message	Structure of the update message The placeholders "%Action%", "%FilesChanged%" and "%UserMessage%" are automatically replaced by the corresponding information during an online change or download. If a placeholder is removed, the corresponding information is removed from the automatically generated comment. Placeholder: "%Action%": indicates whether an online change or download was performed "%FilesChanges%": indicates the number and reason for the changed files (e.g. 2 added). "%UserMessage%": is replaced by the user-specific comment if the option **Ask for update message on each usage** is activated.
Clone	Clones the selected repository to the selected target directory.

6 Reference project settings

6.1 TwinCAT project settings

In order to have the TwinCAT project managed by the multiuser function, this function must be activated in the project settings.



Use Multiuser	Enables the multiuser functionality of the TwinCAT project.
Multiuser URL	Displays the URL of the remote/target system repository.
Init	Initializes the multiuser function for this TwinCAT project.

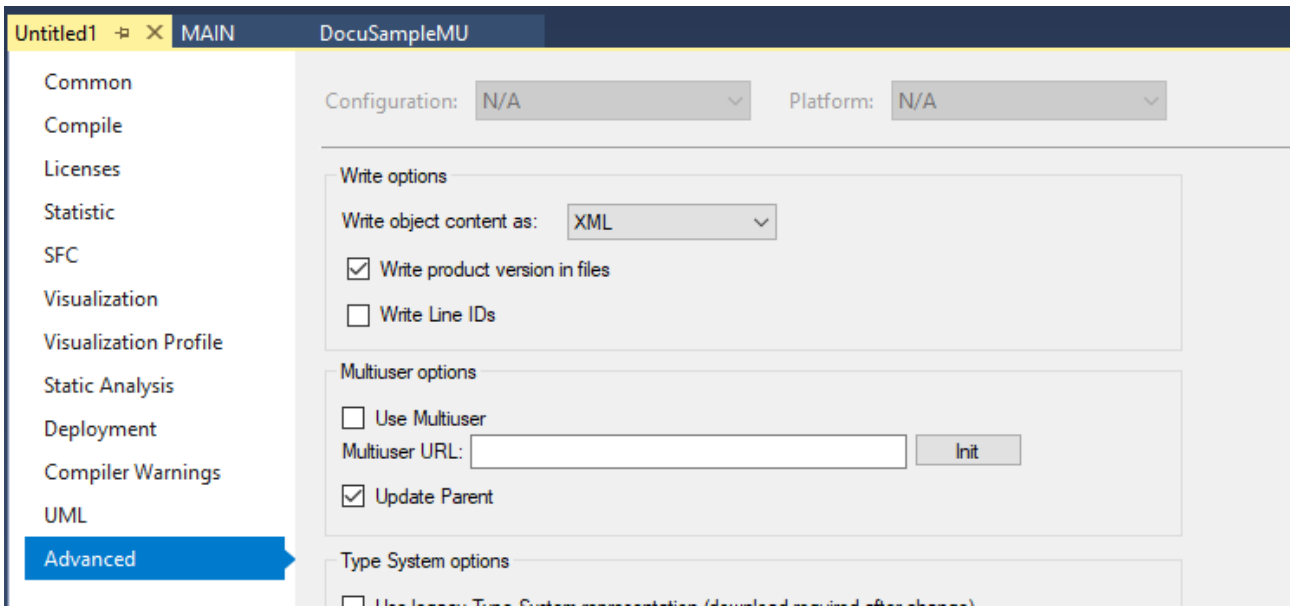
Reuse of PLC projects

The multiuser function is implemented in such a way that in a future version the TwinCAT project and the PLC projects can be managed independently of each other using the multiuser function. Especially in combination with Git as Source Control system during engineering, this offers some advantages if you want to reuse PLC projects in multiple projects. For this reason, it must be defined in the settings of the PLC projects whether they are managed with the overall project or separately.

- Define in the settings of the PLC projects whether they are managed with the overall project or separately. (See PLC project settings [▶ 27])

6.2 PLC project settings

To use the multiuser function only in one PLC project, this function must be enabled in the PLC settings.



Use Multiuser	Enables the multiuser functionality of the PLC project.
Multiuser URL	Displays the URL of the remote/target system repository.
Init	Initializes the multiuser function for this PLC project.
Update Parent	If the entire TwinCAT project is managed by the multiuser function, use this to activate that the TwinCAT project is informed about user activities that require an alignment of the repository (login, download).

Manage TwinCAT project as a whole

The current multiuser function can be used to manage either the entire TwinCAT project or the PLC project. Both are not yet possible, but will follow in a future version. If the entire TwinCAT project is managed, the multiuser function must not be switched on in the PLC, but only the **Update Parent**. In this state the TwinCAT project is informed about corresponding user actions, so that the update of the target system repository takes place from the TwinCAT project.

- Activate the **Update Parent**.

7 Reference server settings

The TwinCAT 3 Multiuser function uses Git. Git itself is separate and is installed optionally, this is selectable during the TwinCAT 3 installation. If a standard Git installation is available on a computer, this can be used for the TwinCAT 3 Multiuser function.

Changing the location of the Git repositories

If necessary, you can change the default storage location of the Git repositories on the target system. The default location is "*C:\ProgramData\Beckhoff\MultiuserRepository*".

To adjust the path, change the entry in the file *"<TwinCAT Folder>/Functions/Multiuser/directorypath.config"* so that it points to the desired path.

8 FAQ

The multiuser function is behaving in a way that is not described. What can I do?

The status of the multiuser function can be viewed in the **Multiuser Explorer** in the **Status** tab. This is the first place that you should look if there is any unexpected behavior when using the multiuser function. (See Information tab [▶ 25])

Also, check whether the TwinCAT 3 AdsGitServer Windows service has been started on the target system and restart it if necessary.

How can I restart a project that has already been created?

Local Git repositories can be deleted in the **Status** tab in the **Multiuser Explorer** using the **DeleteMultiuser files/ Deactive Multiuser** button, which allows you to restart a project that has already been created. (See Information tab [▶ 25])

I'm getting a "RepositoryExistsException" message. How can I resolve this?

A Git repository has already been created for the current project due to an invalid operation in the past. Delete this invalid repository in the **Status** tab in the **Multiuser Explorer** using the **DeleteMultiuser files/ Deactive Multiuser** button.

The History View seems to be messed up, why is this?

Entries are added to the History View in descending order according to their timestamp (with the newest entry at the top). Check whether the times set on the target system and the engineering system are identical.

"Force Update" does not update the multiuser repository on the target system, why is that?

The multiuser function is built on top of the Git source control system. Git always needs to be in a conflict-free state before pushing changes. To this end, the "**Force Update**" function was created, which allows the current local state to be activated on the target system without having to take the Git state into account. This function should only be used if the current state of the local system needs to be run during commissioning without taking the changes made by the other programmers into account. Once the project has been activated, the synchronization of the underlying Git must be performed later on. Start the synchronization either by logging in, reactivating the configuration, or via a manual push. (See Manual tab [▶ 25])

I'm getting an "ADS error 0x745: Timeout has expired" message. How can I resolve this?

In **Multiuser Explorer**, open the **Status** tab. The message "RemoteNotExists" indicates that the remote repository has not been created or has been deleted.

Alternatively, look at the target system, which can also be accessed in the Windows Explorer under *C:\ProgramData\Beckhoff\MultiuserRepository*.

If you get this message several times, restart the TwinCAT 3 AdsGitServer Windows service on the target system.

I'm getting an "ADS error 0x1: An internal error has occurred" message. How can I resolve this?

Check whether the multiuser function has been properly installed. The LibGit2Sharp.dll, TcAdsGitPackage.dll, and TcAdsGitServer.exe files, along with the cmd and mingw32 folders and other elements, must be present in the *C:\TwinCAT\Functions\Multiuser* folder.

9 3rd Party Licenses

The TwinCAT Multiuser functionality uses Git as source control system, which is released under GNU General Public License version 2.0.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies

of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your

freedom to share and change it. By contrast, the GNU General Public

License is intended to guarantee your freedom to share and change free

software--to make sure the software is free for all its users. This

General Public License applies to most of the Free Software

Foundation's software and to any other program whose authors commit to

using it. (Some other Free Software Foundation software is covered by

the GNU Lesser General Public License instead.) You can apply it to

your programs, too.

When we speak of free software, we are referring to freedom, not

price. Our General Public Licenses are designed to make sure that you

have the freedom to distribute copies of free software (and charge for

this service if you wish), that you receive source code or can get it

if you want it, that you can change the software or use pieces of it

in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid

anyone to deny you these rights or to ask you to surrender the rights.

These restrictions translate to certain responsibilities for you if you

distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether

gratis or for a fee, you must give the recipients all the rights that

you have. You must make sure that they, too, receive or can get the

source code. And you must show them these terms so they know their

rights.

We protect your rights with two steps: (1) copyright the software, and

(2) offer you this license which gives you legal permission to copy,

distribute and/or modify the software.
Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software. If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors' reputations.
Finally, any free program is threatened constantly by software
patents. We wish to avoid the danger that redistributors of a free
program will individually obtain patent licenses, in effect making the
program proprietary. To prevent this, we have made it clear that any
patent must be licensed for everyone's free use or not licensed at all.
The precise terms and conditions for copying, distribution and
modification follow.
GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION
0. This License applies to any program or other work which contains
a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License. The "Program", below,
refers to any such program or work, and a "work based on the Program"
means either the Program or any derivative work under copyright law:
that is to say, a work containing the Program or a portion of it,
either verbatim or with modifications and/or translated into another
language. (Hereinafter, translation is included without limitation in
the term "modification".) Each licensee is addressed as "you".
Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running the Program is not restricted, and the output from the Program
is covered only if its contents constitute a work based on the
Program (independent of having been made by running the Program).
Whether that is true depends on what the Program does.
1. You may copy and distribute verbatim copies of the Program's
source code as you receive it, in any medium, provided that you
conspicuously and appropriately publish on each copy an appropriate
copyright notice and disclaimer of warranty; keep intact all the
notices that refer to this License and to the absence of any warranty;

and give any other recipients of the Program a copy of this License along with the Program.
You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.
2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)
These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.
Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or

collective works based on the Program.
In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.
3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)
The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.
If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not

compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under

any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free

Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by

the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
Also add information on how to contact you by electronic and paper mail.
If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.
You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License.

10 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

Trademark statements

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, 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.

Third-party trademark statements

Microsoft, Microsoft Azure, Microsoft Edge, PowerShell, Visual Studio, Windows and Xbox are trademarks of the Microsoft group of companies.

More Information:
www.beckhoff.com

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