TwinCAT 3.1 Build 4024: More efficiency and consistency in the engineering process

# New TwinCAT properties reduce engineering costs and increase communication stability

Every two years, Beckhoff releases a new build of TwinCAT. Each build contains many new features that make working with TwinCAT even more efficient. The latest build 4024 is no exception. It again improves efficiency and consistency in the engineering process, which reduces costs. In addition, it enhances communication stability with Secure ADS. It also integrates the latest versions of Visual Studio®.

The new Build 4024 of TwinCAT 3.1 replaces the Visual Studio® 2013 shell with the new advanced and improved Visual Studio® 2017 shell. As a result, the start page of Visual Studio® now contains current information individualized for TwinCAT users that is imported into Visual Studio® via an RSS feed, which can also be subscribed to separately.

**Multi-user access to the PLC**

Commissioning new systems is becoming more expensive all the time. In addition, getting qualified personnel for a major commissioning job is often difficult. Accordingly, there is a strong desire for improved capabilities to commission even complex systems quickly and easily. However, this is only possible if you can enable multiple programmers to work simultaneously. With its new multi-user capability, TwinCAT now offers exactly the right solution for this challenge. For this new function Beckhoff employs a principle that is already being used for offline development projects by teams. Naturally, the only way to develop software offline collaboratively is with a source code control system. The new TwinCAT build features this capability and has implemented it for online operation in a way that requires no additional knowledge.

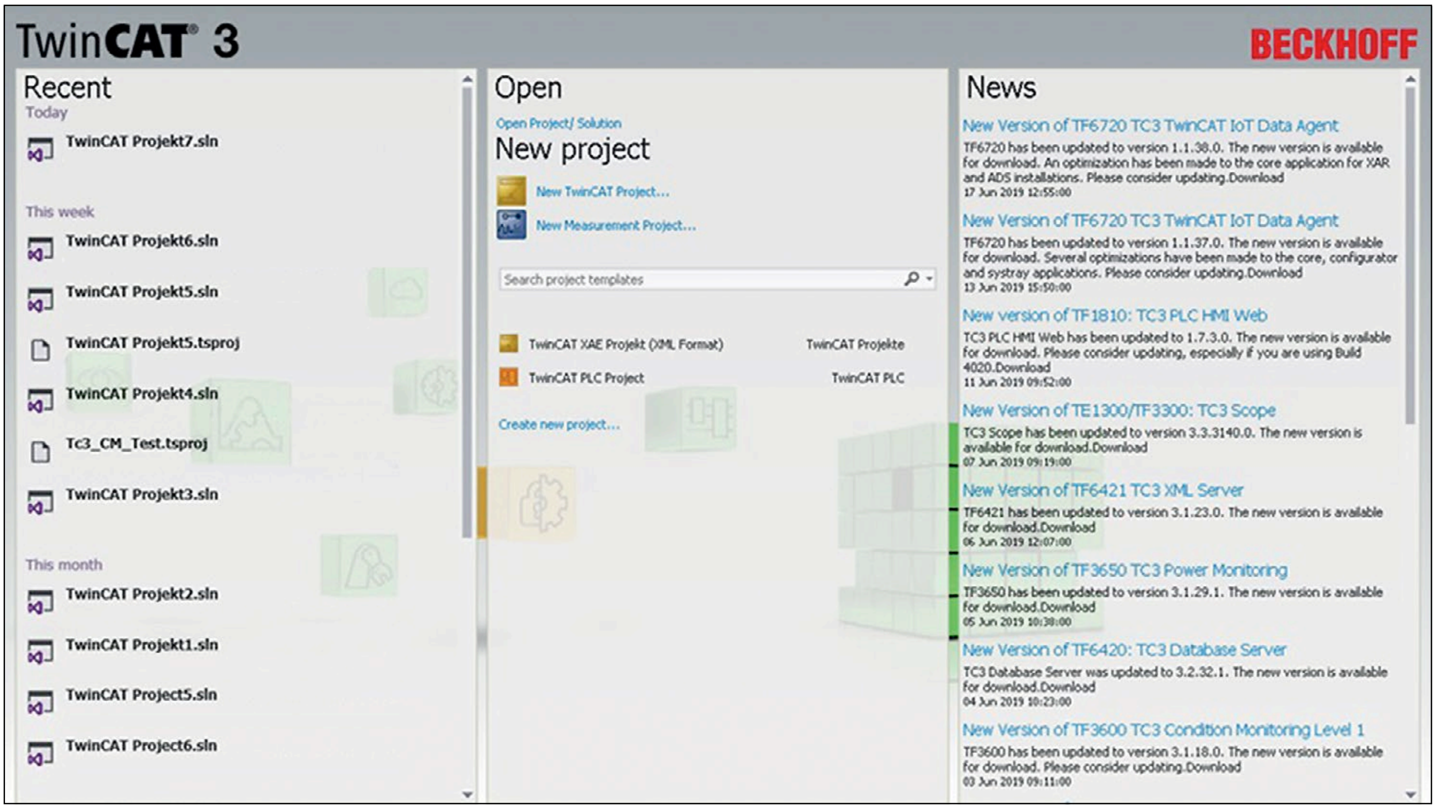
What does this kind of online teamwork on a control application look like? Let's assume that programmer 1 logs onto the controller and makes a change in the code while programmers 2 and 3 are also logged in. Since programmers 2 and 3 don't have the source code of the change on their programming devices, the development system must store programmer 1's source code in a way that grants programmers 2 and 3 access to it. TwinCAT does this with Git, the popular open-source solution for source code control, and a repository that can be installed directly on the controller. As soon as programmer 2 logs in, a window opens that displays the changes from the source code status on programmer 2's development computer. Programmer 2 can then accept the changes made by programmer 1 or combine both versions with the familiar TwinCAT Compare Tool. Programmer 2 sees the differences between the source code versions and must decide which one to accept. Another advantage of using Git on the controller is that the user can check at all times who made which change. This ensures seamless tracking capability during the commissioning process as well as during the entire life cycle of the machine.

**Variant management avoids errors**

If several variants of a machine are to be mapped in software, up to now either a different source code version had to be used for each variant or a meta software application had to be developed that covers all variants. The specifics of a variant were then generated via parameterization. Both methods are cumbersome and error-prone:

– Different software applications are difficult to maintain. In addition, any bug fixes must be performed in all software variants.
– Meta software has the disadvantage that more code than necessary is visible and the configuration shows the maximum expansion, most of which is not being used for a particular version, however. Here, too, maintaining the software is difficult.

Multi-user access to PLC
Time-synchronized datasets in distributed installations
Secure communication with Secure ADS
TwinCAT® 3



Recent

Today

TwinCAT Projekt7.sln

This week

TwinCAT Projekt6.sln

TwinCAT Projekt5.sln

TwinCAT Projekt5.tsproj

TwinCAT Projekt4.sln

Tc3_CM_Test.tsproj

TwinCAT Projekt3.sln

This month

TwinCAT Projekt2.sln

TwinCAT Projekt1.sln

TwinCAT Project5.sln

TwinCAT Project6.sln

Open

Open Project/ Solution

New project

New TwinCAT Project...

New Measurement Project...

Search project templates

| | |
|---|---|
| TwinCAT XAE Projekt (XML Format) | TwinCAT Projekte |
| TwinCAT PLC Project | TwinCAT PLC |

Create new project...

News

New Version of TF6720 TC3 TwinCAT IoT Data Agent

TF6720 has been updated to version 1.1.38.0. The new version is available for download. An optimization has been made to the core application for XAR and ADS installations. Please consider updating.Download
17 Jun 2019 12:55:00

New Version of TF6720 TC3 TwinCAT IoT Data Agent

TF6720 has been updated to version 1.1.37.0. The new version is available for download. Several optimizations have been made to the core, configurator and systray applications. Please consider updating.Download
13 Jun 2019 15:50:00

New version of TF1810: TC3 PLC HMI Web

TC3 PLC HMI Web has been updated to 1.7.3.0. The new version is available for download. Please consider updating, especially if you are using Build 4020.Download
11 Jun 2019 09:52:00

New Version of TE1300/TF3300: TC3 Scope

TC3 Scope has been updated to version 3.3.3140.0. The new version is available for download.Download
07 Jun 2019 09:19:00

New Version of TF6421 TC3 XML Server

TF6421 has been updated to version 3.1.23.0. The new version is available for download.Download
06 Jun 2019 12:07:00

New Version of TF3650 TC3 Power Monitoring

TF3650 has been updated to version 3.1.29.1. The new version is available for download.Download
05 Jun 2019 10:38:00

New Version of TF6420: TC3 Database Server

TC3 Database Server was updated to 3.2.32.1. The new version is available for download.Download
04 Jun 2019 10:23:00

New Version of TF3600 TC3 Condition Monitoring Level 1

TF3600 has been updated to version 3.1.18.0. The new version is available for download. Please consider updating.Download
03 Jun 2019 09:11:00

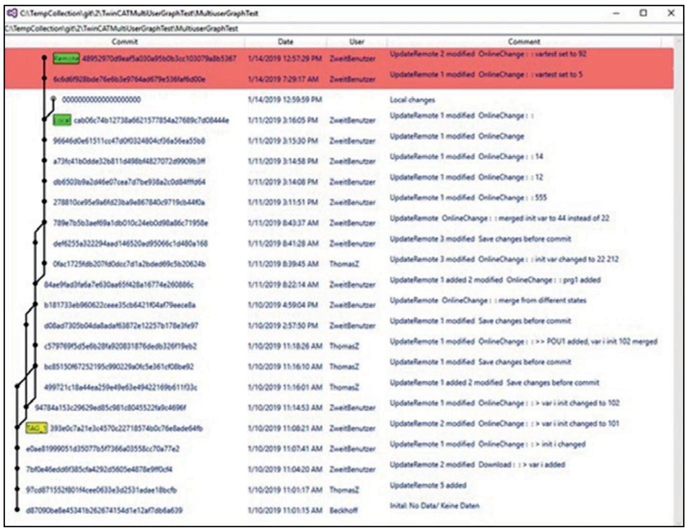
**TwinCAT users now receive the latest information in a targeted manner via the Visual Studio® start page.**

The optimal solution is to have software that is generated in different variants based on simple settings. In classical high-level language programming, conditional compiling is used for this purpose. A TwinCAT project, however, involves not only PLC programming but also variants with different I/O equipment. Accordingly, a device may not always be present or it may have different parameters. The same applies in the area of axis motion control. For these reasons, the new TwinCAT build has so-called project variants as well as variant groups for the entire TwinCAT project. That way, variant-specific settings can be made in areas such as I/O or motion configuration. They also make maintaining the software a lot easier, because there is only one program for all machine variants. To generate a specific variant, the user only needs to select it, and the correct program is generated automatically. Versioning and maintaining the software becomes very easy in this way.

## Improvements in the PLC area

To develop and test PLC code faster and more effectively, many improvements have also been incorporated in the PLC area. Especially in object-oriented programming, simplifying code maintenance and reusability is a major objective. The use of interfaces is essential in this context. In the past, interface pointers were able to indicate the address, but no pointer de-referencing was executed. With Build 4024, this is now possible and further improves the engineering capabilities.

In the C++ and C# high-level languages, programming abstract classes with abstract methods and properties is widespread. This option is now also available to the PLC programmer, because abstract classes make it easy to generate sample function blocks or classes, which can then be filled with code. This ability brings the PLC programmer another step closer to the capabilities of C++ and C#. The same applies to exception handling. If there is an exception, the PLC programmer wants to respond to it. In case of a division by zero, it may be useful to execute a specific routine to bring the machine into a safe state and prevent any damage to it.



C:\TempCollection\git\2\TwinCATMultiUserGraphTest\MultiuserGraphTest

C:\TempCollection\git\2\TwinCATMultiUserGraphTest\MultiuserGraphTest

| Commit | Date | User | Comment |
|---|---|---|---|
| Remote 48952970d9eaf5a030a95b0b3cc103079a8b5367 | 1/14/2019 12:57:29 PM | ZweitBenutzer | UpdateRemote 2 modified OnlineChange : : vartest set to 92 |
| 6c6d6f928bde76e6b3e9764ad679e536faf6d00e | 1/14/2019 7:29:17 AM | ZweitBenutzer | UpdateRemote 1 modified OnlineChange : : vartest set to 5 |
| 0000000000000000000 | 1/14/2019 12:59:59 PM | | Local changes |
| Local cab06c74b12738a6621577854a27689c7d08444e | 1/11/2019 3:16:05 PM | ZweitBenutzer | UpdateRemote 1 modified OnlineChange : : |
| 96646d0e61511cc47d0f0324804cf36a56ea55b8 | 1/11/2019 3:15:30 PM | ZweitBenutzer | UpdateRemote 1 modified OnlineChange |
| a73fc41b0dde32b811d498bf4827072d9909b3ff | 1/11/2019 3:14:58 PM | ZweitBenutzer | UpdateRemote 1 modified OnlineChange : : 14 |
| db6503b9a2d46e07cea7d7be938a2c0d84ffd64 | 1/11/2019 3:14:08 PM | ZweitBenutzer | UpdateRemote 1 modified OnlineChange : : 12 |
| 278810ce95e9a6fd23ba9e867840c9719cb44f0a | 1/11/2019 3:11:51 PM | ZweitBenutzer | UpdateRemote 1 modified OnlineChange : : 555 |
| 789e7b5b3aef69a1db010c24eb0d98a86c71958e | 1/11/2019 8:43:37 AM | ZweitBenutzer | UpdateRemote OnlineChange : : merged init var to 44 instead of 22 |
| def6255a322294aad146520ed95066c1d480a168 | 1/11/2019 8:41:28 AM | ZweitBenutzer | UpdateRemote 3 modified Save changes before commit |
| 0fac1725fdb207fd0dcc7d1a2bded69c5b20624b | 1/11/2019 8:39:45 AM | ThomasZ | UpdateRemote 3 modified OnlineChange : : init var changed to 22 212 |
| 84ae9fad3fa6a7e630aa65f428a16774e260886c | 1/11/2019 8:22:14 AM | ZweitBenutzer | UpdateRemote 1 added 2 modified OnlineChange : : prg1 added |
| b181733eb960622ceee35cb6421f04af79eece8a | 1/10/2019 4:59:04 PM | ZweitBenutzer | UpdateRemote OnlineChange : : merge from different states |
| d08ad7305b04da8adaf63872e12257b178e3fe97 | 1/10/2019 2:57:50 PM | ZweitBenutzer | UpdateRemote 1 modified Save changes before commit |
| c579769f5d5e6b28fa920831876dedb326f19eb2 | 1/10/2019 11:18:26 AM | ThomasZ | UpdateRemote 1 modified OnlineChange : : >> POU1 added, var i init 102 merged |
| bc85150f67252195c990229a0fc5e361cf08be92 | 1/10/2019 11:16:10 AM | ThomasZ | UpdateRemote 1 modified Save changes before commit |
| 499721c18a44ea259e49e63e49422169b611f33c | 1/10/2019 11:16:01 AM | ThomasZ | UpdateRemote 1 added 2 modified Save changes before commit |
| 94784a153c29629ed85c981c8045522fa9c4696f | 1/10/2019 11:14:53 AM | ZweitBenutzer | UpdateRemote 1 modified OnlineChange : : > var i init changed to 102 |
| TAG_1 393e0c7a21e3c4570c22718574b0c76e8ade64fb | 1/10/2019 11:08:21 AM | ZweitBenutzer | UpdateRemote 2 modified OnlineChange : : > var i init changed to 101 |
| e0ae81999051d35077b5f7366a03558cc70a77e2 | 1/10/2019 11:07:41 AM | ZweitBenutzer | UpdateRemote 1 modified OnlineChange : : > init i changed |
| 7bf0e46edd6f385cfa4292d5605e4878e9ff0cf4 | 1/10/2019 11:04:20 AM | ZweitBenutzer | UpdateRemote 2 modified Download : : > var i added |
| 97cd871552f801f4cee0633e3d2531adae18bcfb | 1/10/2019 11:01:17 AM | ThomasZ | UpdateRemote 5 added |
| d87090be8e45341b262674154d1e12af7db6a639 | 1/10/2019 11:01:15 AM | Beckhoff | Initial: No Data/ Keine Daten |

**The multi-user functionality of TwinCAT simplifies the commissioning of even complex machines significantly.**

The mapping dialog has been improved in such a way that not all node levels are open. This reduces the scrolling range and makes it easier to create a link, for example between the PLC and an I/O component, because the respective devices can now be found more rapidly. An option allows you to switch between the old, the new, and an automatically adjusted view. With the "Go To Link" variable, users have always been able to jump back and forth between an I/O device and the process image, but there was no way to jump from the process image to the code. This is now possible with the "Go To Definition" command, a feature which makes debugging much easier. It also improves program clarity, because with this functionality the still frequently used lists of global variables can be dispensed with.

Another advantage is that the program code of function blocks in graphical languages can now be saved in the binary Base65 format as well, which speeds up the loading and saving of programs with a large proportion of graphical languages.

## Online change of C++- or MATLAB®/Simulink® code

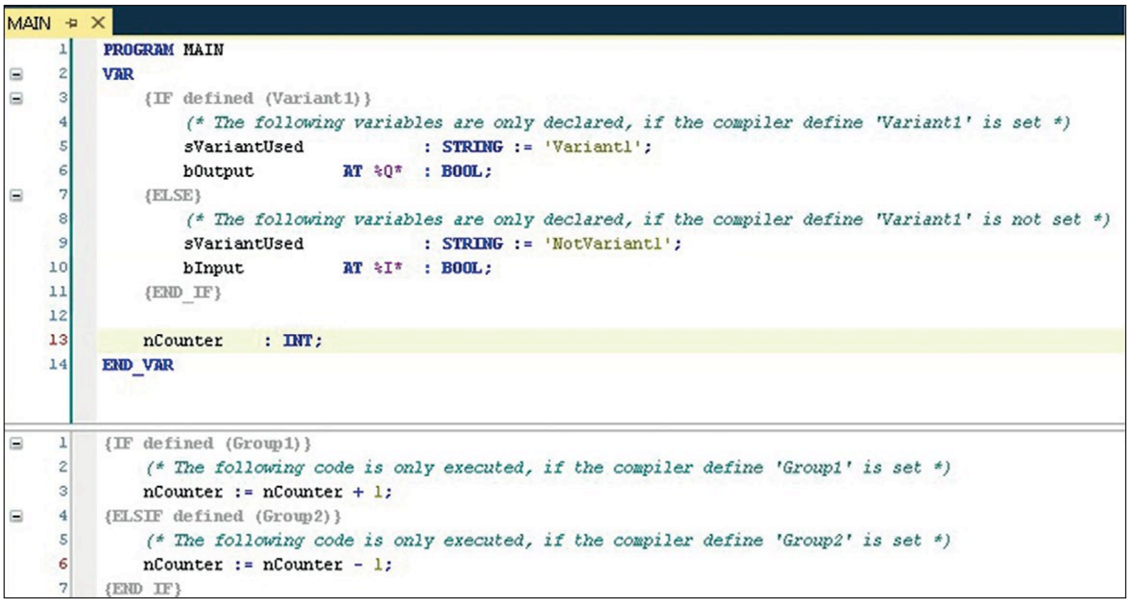
Until now, only the PLC programmer was able to modify the control code during ongoing machine operation. Since many machines or production systems are stopped rarely or not at all, this is a critical feature. If, however, you use C++ or MATLAB®/Simulink® to control the machine, up to now you could only modify this code in connection with a restart of the machine.

Unfortunately, such restarts are not permissible in some applications. Many processes must not be interrupted, especially in the field of test and inspection machines, while program modifications are frequently needed. To resolve this problem, Beckhoff has implemented in the new TwinCAT build the ability to replace C++ and MATLAB®/Simulink® code while the machine or system is running. To do this, the code is given a special version so that one version can be replaced with a new one on the fly without losing any data.

## Time-synchronized datasets in distributed installations

In larger installations with multiple controllers, certain data must often be stored on a central server so that the aggregated data can be subsequently analyzed. Unfortunately, the timestamps on the different devices cannot be generated uniformly. If such data is then aggregated for a centralized application like TwinCAT Scope, the central system has no accurate time reference. Unfortunately, the system clocks on the individual controllers cannot be simply adjusted to synchronize them, because this would impair the entire application. The only option that's left is therefore to use a reference clock and determine its offsets from the local clocks when timestamps are generated for the data to be aggregated.

But which reference clock should you use? In order to achieve medium accuracy very simply, you can use the (S)NTP service, which is free and installed on all Beckhoff devices. In distributed installations, this method is sufficient when all you want to do is compare data. For more accuracy, a process based on IEEE 1588 can be used. This method, which is also called PTP (Precision Time Protocol), can reconcile very different PLCs with great precision, but requires special hardware such as the EL6688 EtherCAT Terminal. The only way to achieve even



```
PROGRAM MAIN
VAR
    {IF defined (Variant1)}
        (* The following variables are only declared, if the compiler define 'Variant1' is set *)
        sVariantUsed            : STRING := 'Variant1';
        bOutput         AT %Q*  : BOOL;
    {ELSE}
        (* The following variables are only declared, if the compiler define 'Variant1' is not set *)
        sVariantUsed            : STRING := 'NotVariant1';
        bInput          AT %I*  : BOOL;
    {END_IF}

    nCounter    : INT;
END_VAR
```

```
{IF defined (Group1)}
    (* The following code is only executed, if the compiler define 'Group1' is set *)
    nCounter := nCounter + 1;
{ELSIF defined (Group2)}
    (* The following code is only executed, if the compiler define 'Group2' is set *)
    nCounter := nCounter - 1;
{END_IF}
```

**The new variant management capability of TwinCAT simplifies the versioning and maintenance of control software significantly.**

more clock precision would be with special cabling, i.e. to use an EtherCAT cable between the devices. The interface introduced in the new TwinCAT build for one of these methods allows you to use corrected timestamps for different data and thus achieve comparable timestamps in aggregated databases.

**Secure communication with Secure ADS**

The Beckhoff ADS protocol was already introduced with the first version of TwinCAT and has been essentially unchanged ever since, but frequently enhanced by new features all the same. One such new feature in TwinCAT Build 4024 is the ability to communicate securely via ADS by setting up an encrypted connection between two parties who then exchange ADS telegrams as usual.

When a new connection is created between two ADS devices, their authentication can be ensured via certificates. All telegrams are automatically encrypted. Beckhoff has decided to use TLS, the best-known process for authentication and encryption, for this purpose. The programmer does not need to worry about the certificates and their management, because TwinCAT handles this automatically in the background. The integration into the central communication component, TwinCAT Router, also enables legacy applications using ADS to use encrypted connections by reconfiguring them without having to recompile or even rewrite the application.

**New features in safety engineering**

In addition to security, safety plays an important role in the TwinCAT system as well. For this reason, numerous features have been added that make safety-related operations easier. One of the most important of these is the multiple use of variables. Variables therefore no longer need to be unique but can be declared and used locally as well as globally across an entire safety project. This makes safety-related projects much easier and more effective.

In addition, special function blocks that are based on existing and certified function blocks can be defined and instantiated multiple times. This means that a functionality for a safety door e.g. needs to be created only once. If an application has multiple safety doors, the function block can simply be reused in multiple instances and with different I/O assignments or parameters.

**Dr. Josef Papenfort, Senior Product Manager TwinCAT, Beckhoff Automation**

More information:
www.beckhoff.com/twincat3