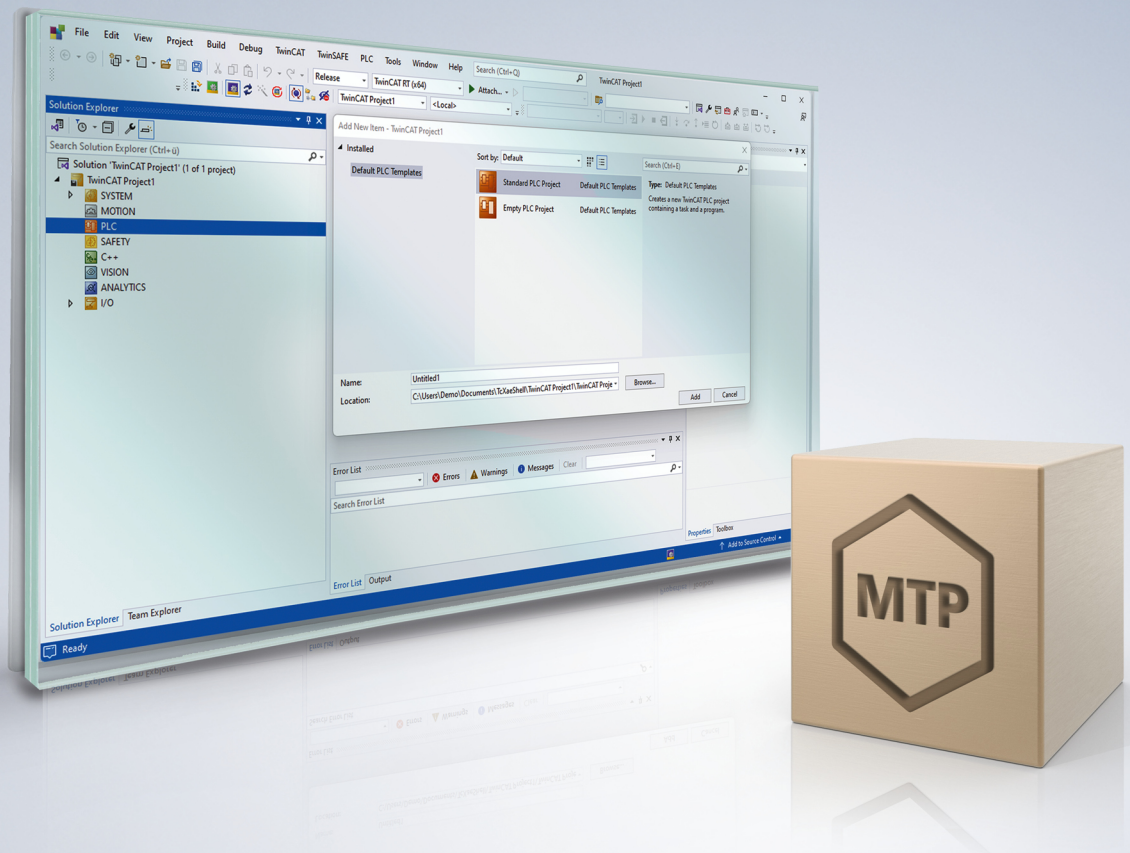
BECKHOFF New Automation Technology

Manual | EN

TE8400

TwinCAT 3 | MTP Engineering



2025-11-04 | Version: 1.0.0

Table of contents

1 Foreword 5
1.1 Notes on the documentation 5
1.2 For your safety 5
1.3 Notes on information security 7
2 Introduction 8
3 Installation 9
4 Quickstart 10
4.1 Creating a new MTP project 10
4.2 Creating a new MTP module 12
4.3 Overview MTP Engineering 13
4.4 Creating a new HMI picture 14
4.5 Creating a VisualObject 15
4.6 Creating new DataObjects 17
4.7 MTP export 18
4.8 Generating PLC and HMI projects 18
4.9 Automatic PLC code generation 20
4.10 HMI generation 23

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®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar®, and XTS® 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 Introduction

An MTP (Module Type Package) is a communication interface between a higher-level control system (e.g. process control system) and a modular plant. The MTP describes the interface of visualization, provided services and other features of a module, so that a simple orchestration of the modules to form a complete plant can be done easily and quickly in the higher-level control system.

The MTP concept is integrated in TwinCAT 3 as a software solution by the TF8400 TwinCAT 3 MTP Runtime and TE8400 TwinCAT 3 MTP Engineering products. In this way, the communication between the higher-level control system (POL) and the individual modules of a complex plant can be described.

The features and services that a software-based plant module should have and the dependencies between the defined services are defined in the MTP Engineering. There is a high degree of individualization for ideal adaptation to the processes on site.

Through a separate project type, the MTP-specific project planning, such as the creation of services or parameters, is possible in a separate MTP folder. Through the common solution, TwinCAT 3 Engineering offers the possibility to create, modify and manage the MTP as well as the source code of the PLC in one environment. The connection of TwinCAT 3 Engineering to version control enables consistent and sustainable storage of a project, including PLC source code, MTP description and other program sources.

3 Installation

System requirements

Technical Data	Requirements
Operating system	Windows 10, 11
Target platform	PC architecture (x64)
TwinCAT version	TwinCAT 3.1 Build 4026
Required TwinCAT license	TE8400 TwinCAT 3 MTP Engineering

TwinCAT Package Manager: Installation (TwinCAT 3.1 Build 4026)

Detailed instructions on installing products can be found in the chapter Installing workloads in the TwinCAT 3.1 Build 4026 installation instructions.

Install the following workload to be able to use the product:

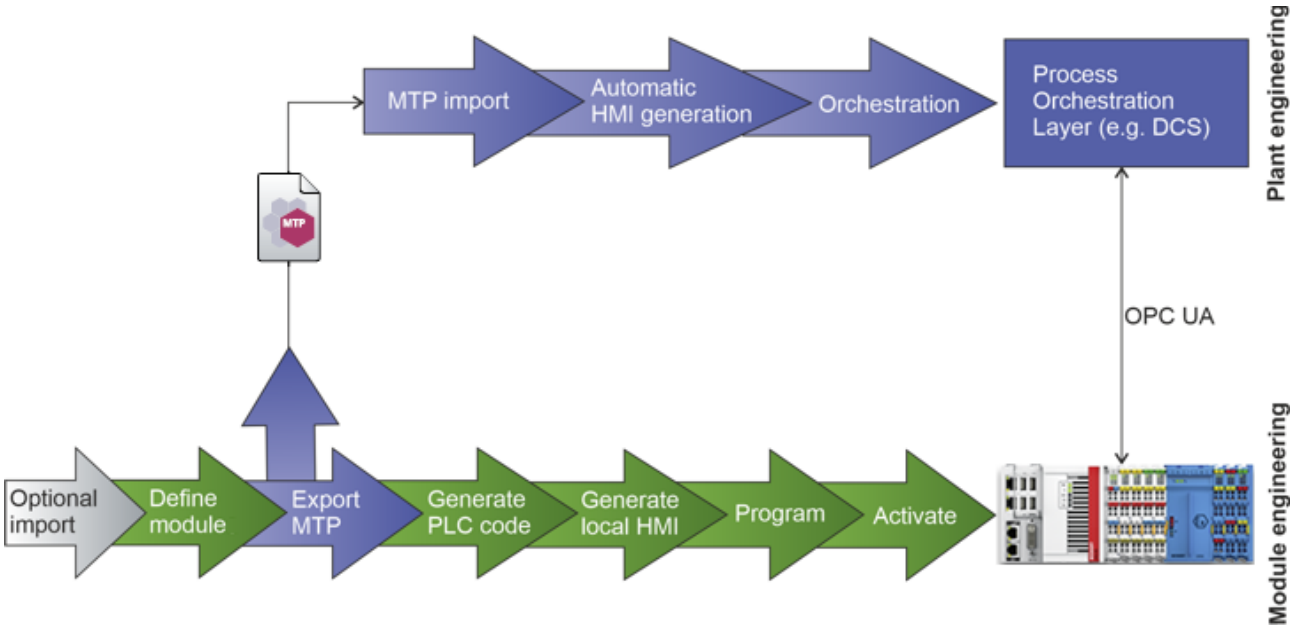
TE8400 | TwinCAT 3 MTP Engineering

4 Quickstart

These instructions are intended to provide an overview of the structure and working with TwinCAT 3 MTP Engineering. It mainly deals with standard setting options. The following topics are considered:

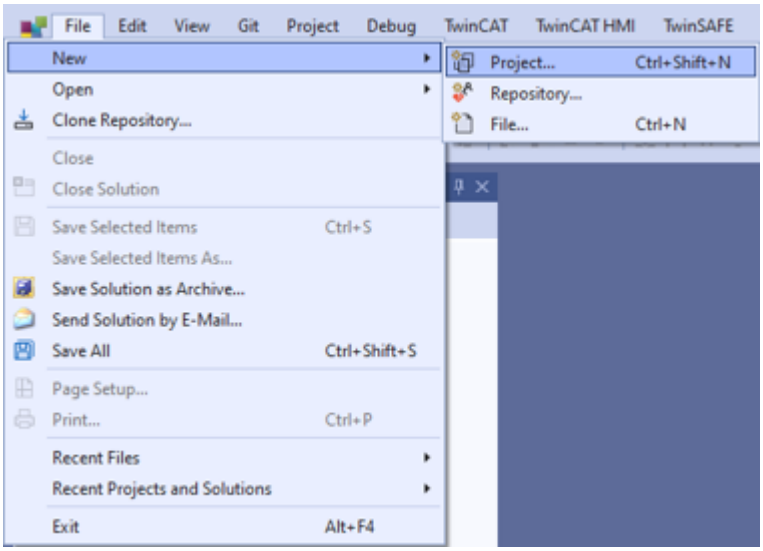
1. Creating a new MTP project [▶ 10] with an MTP module [▶ 12] in TwinCAT
2. Explanation of the MTP Engineering workspace [▶ 13]
3. Creating an MTP HMI image [▶ 14]
4. Creating MTP objects [▶ 15]
5. MTP export [▶ 18]
6. PLC code generation [▶ 20]
7. HMI generation [▶ 23]

The following graphic shows the Beckhoff MTP workflow:

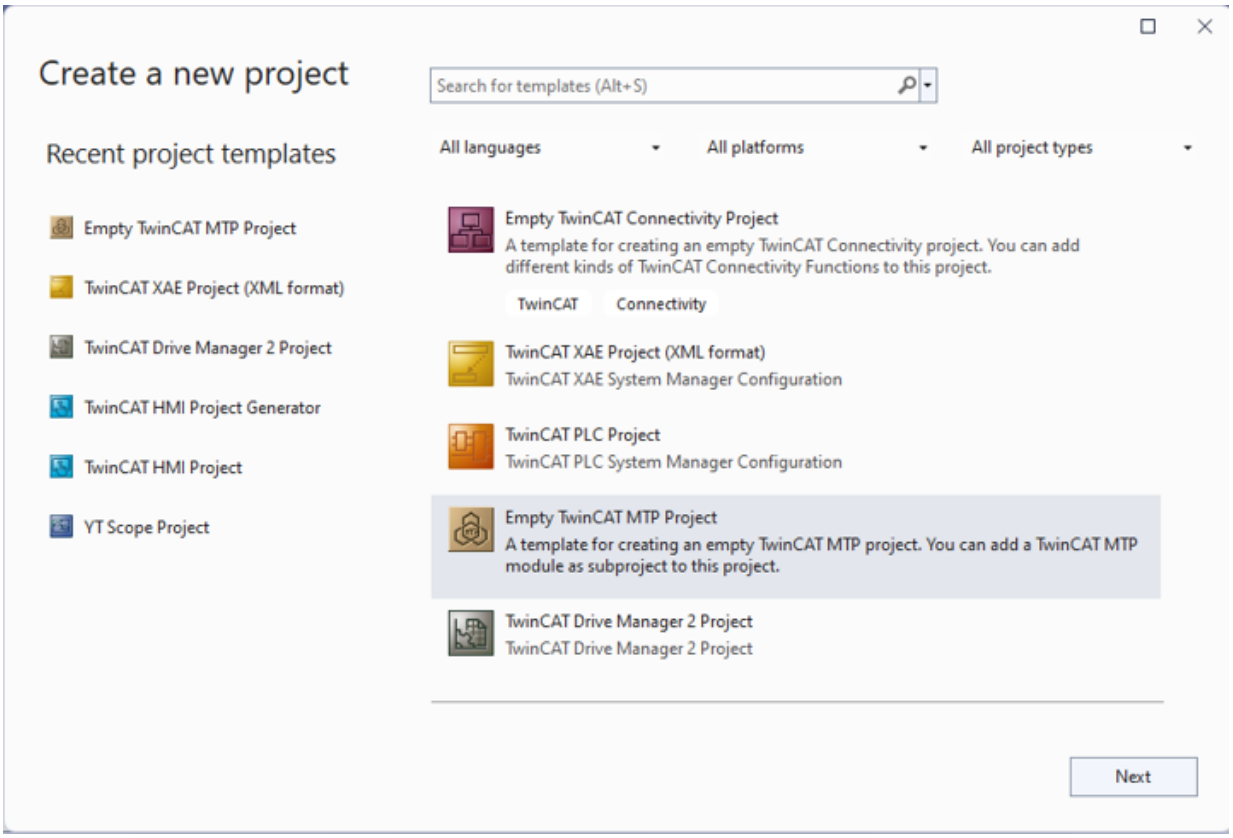


4.1 Creating a new MTP project

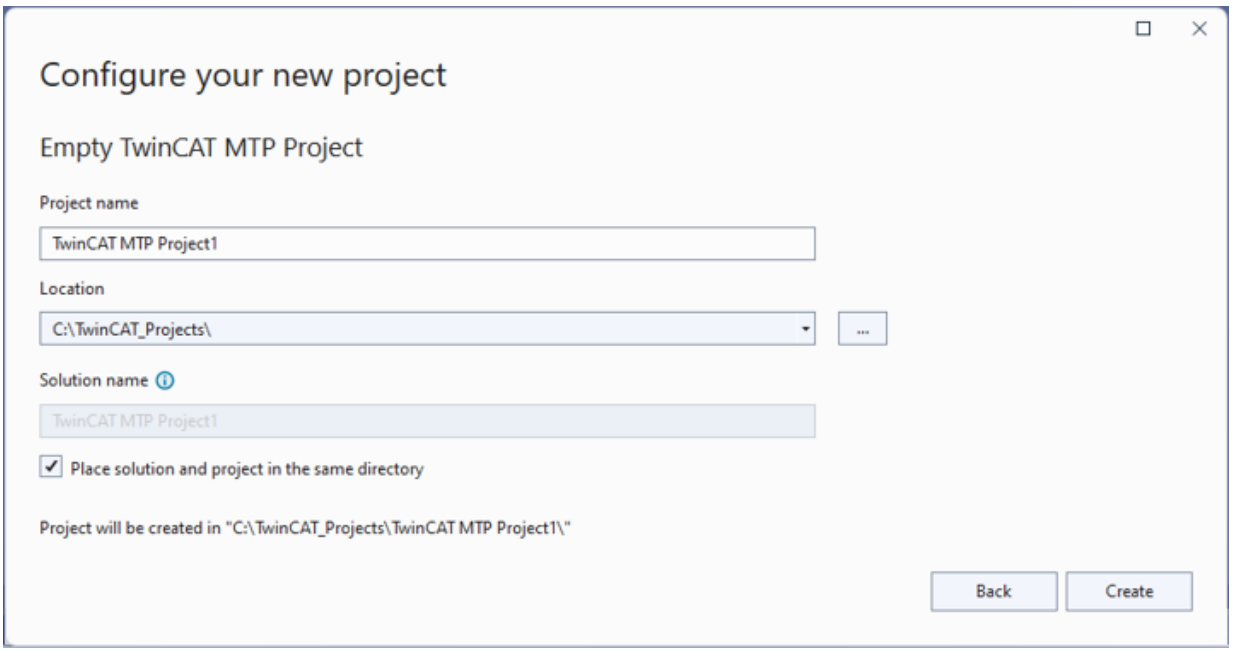
1. Under **File->New**, click on **Project**.



2. Select the project type **Empty TwinCAT MTP Project**.



3. Click on **Next**.
4. Assign a project name and select a storage location.



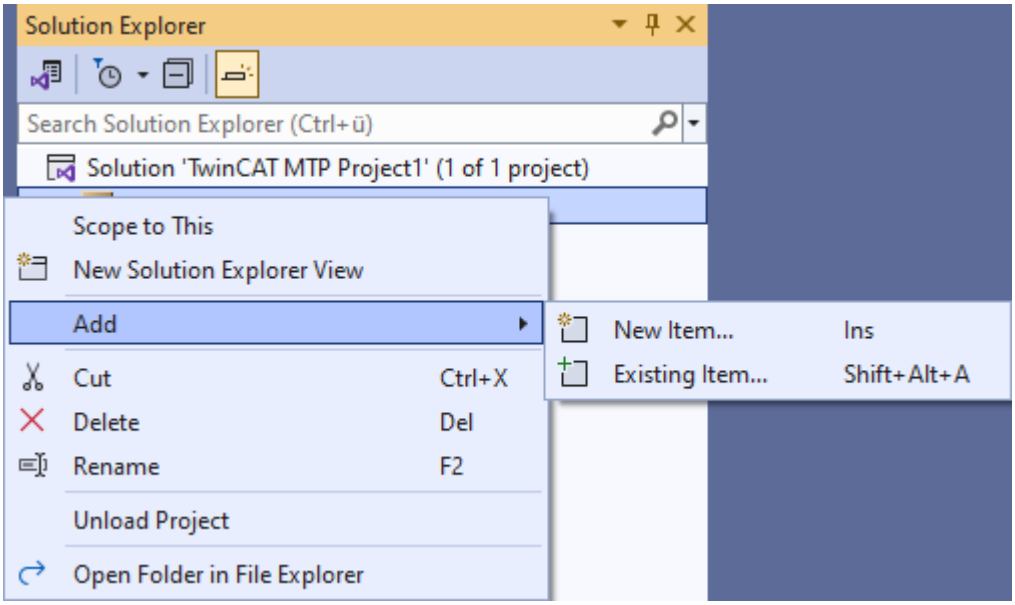
5. Click on **Create** to create the new MTP project.

⇨ The MTP project is created.

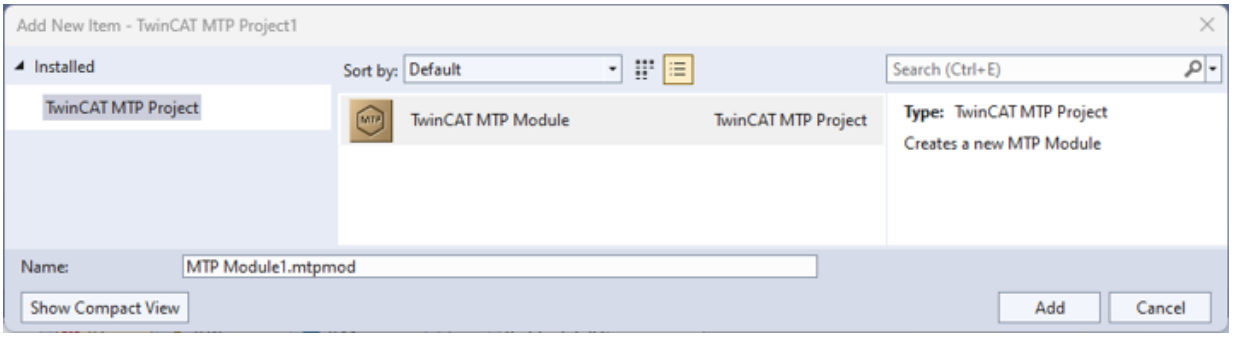
4.2 Creating a new MTP module

You can create one or more MTP modules in the MTP project.

1. In the context menu of the MTP project, select **Add->New Item**.



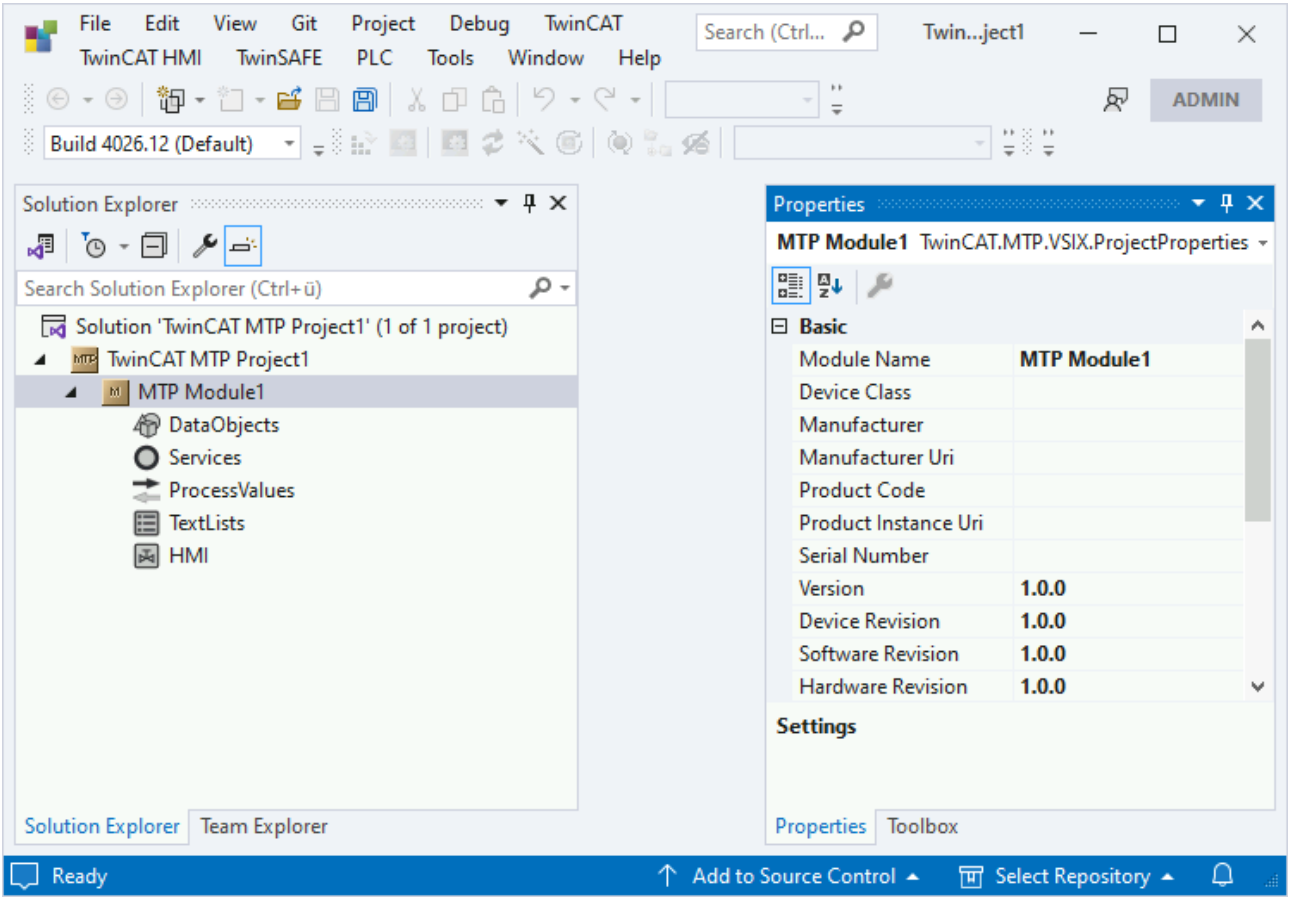
2. Click on **Add** to create an empty MTP module.



⇨ The empty MTP module is created.

4.3 Overview MTP Engineering

Once you have created the new MTP module, the following project structure appears in the Solution Explorer.



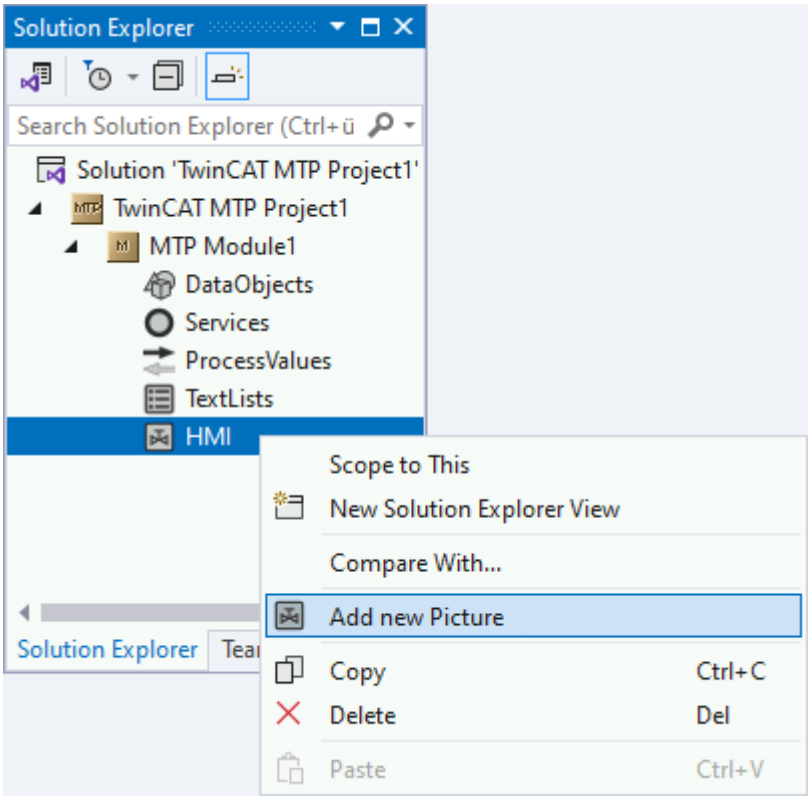
New objects, HMI pages and services can be created and displayed in the Solution Explorer. These are divided into the following aspects:

- `DataObjects`: List of MTP-specific data objects such as an analog drive `AnaDrv`. An FB instance is later created in the PLC for each of these elements.
- `Services`: List of services with their associated procedures, parameters and report values. FB instances are later created in the PLC for the services and their components.
- `ProcessValues`: List of cross-module incoming and outgoing process values. An FB instance is later created in the PLC for each of these elements.
- `TextLists`: List of project-related and service-related enumerations. These elements are later transferred as objects to the HMI and as enumerations to the PLC.
- `HMI`: List of HMI pictures that can later be converted into TwinCAT HMI pages (`.content`).

You can generally fill the above aspects independently of each other. We recommend starting with the HMI.

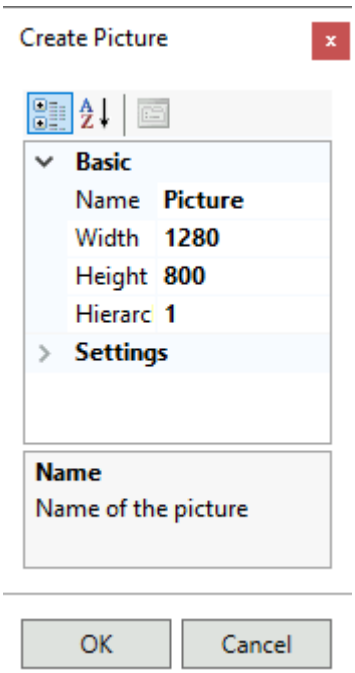
4.4 Creating a new HMI picture

1. Open the context menu of **HMI** in the Solution Explorer and select **Add new Picture** to create a new HMI picture.



⇨ A dialog opens.

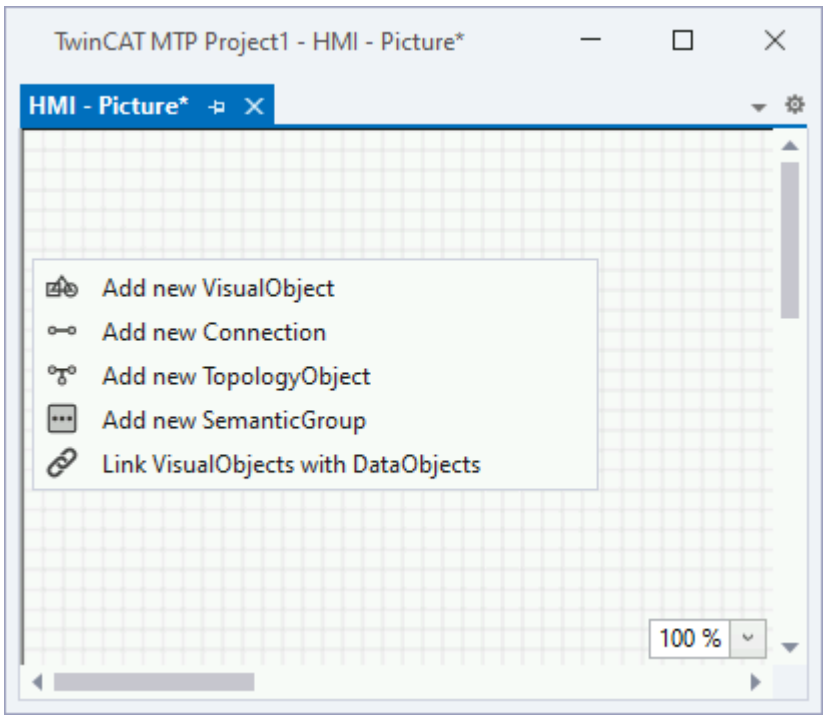
2. Change the most important properties under **Basic**.



3. Confirm your settings with **OK**.

⇨ The newly created HMI picture opens.

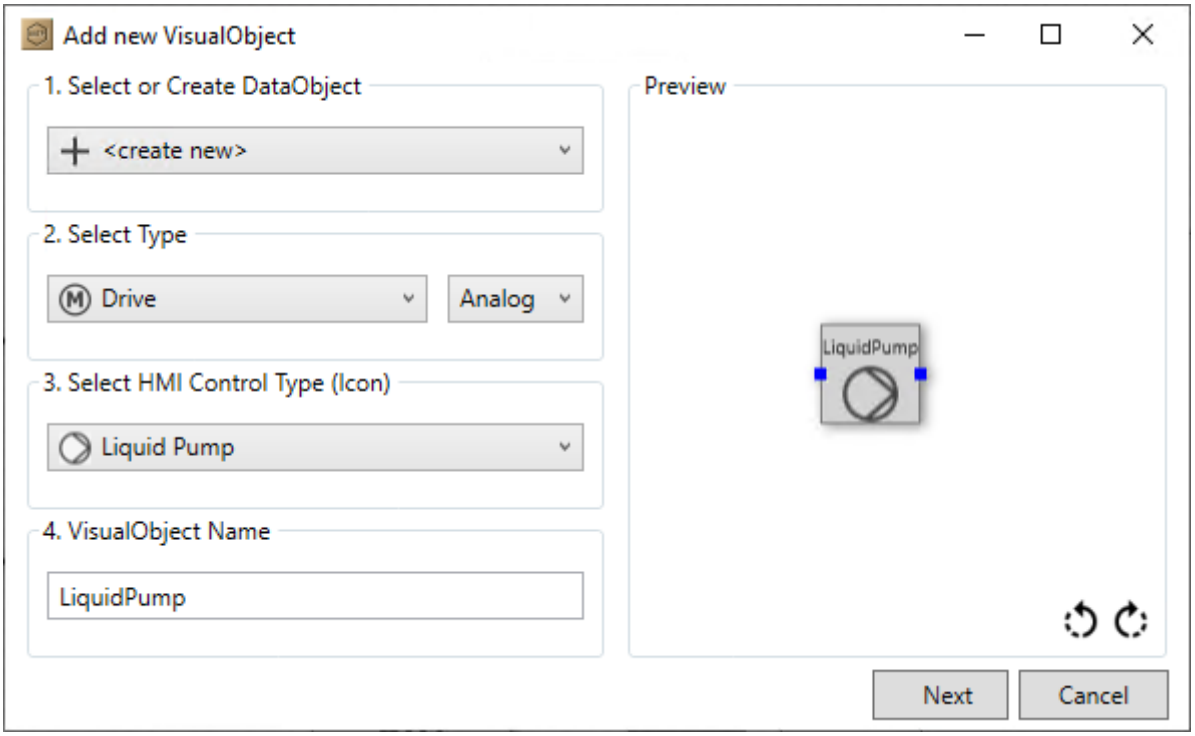
The workspace includes a grid by default. In the context menu of the picture, you have the option to create a new VisualObject.



4.5 Creating a VisualObject

A `VisualObject` can be static or dynamic. A static `VisualObject` can be considered a simple graphic. A dynamic `VisualObject` is linked to a `DataObject` and thus represents a link between the HMI and the PLC.

When you select **Add new VisualObject** in the context menu of the HMI picture, a dialog opens:



When creating a `VisualObject`, you can make the following settings in the **Add new VisualObject** dialog:

1. **Select or Create** `DataObject`
   - New `DataObject`, that will be linked to the `VisualObject`.
   - Link to an existing `DataObject`

- No link to a `DataObject`

2. **Select** `DataObject` **Type**
   - When you create a new `DataObject`, the `Select Type` field is displayed and you can select the type of `DataObject` here.
3. **Select** `Icons`
   - Depending on the selected `DataObject`, an icon matching the `DataObject` is preselected. You can customize the icon via the drop-down menu if necessary.
4. `VisualObject` **Name**
   - If you have selected an existing `DataObject`, the name will be taken from `DataObject`. You can adjust this if necessary.

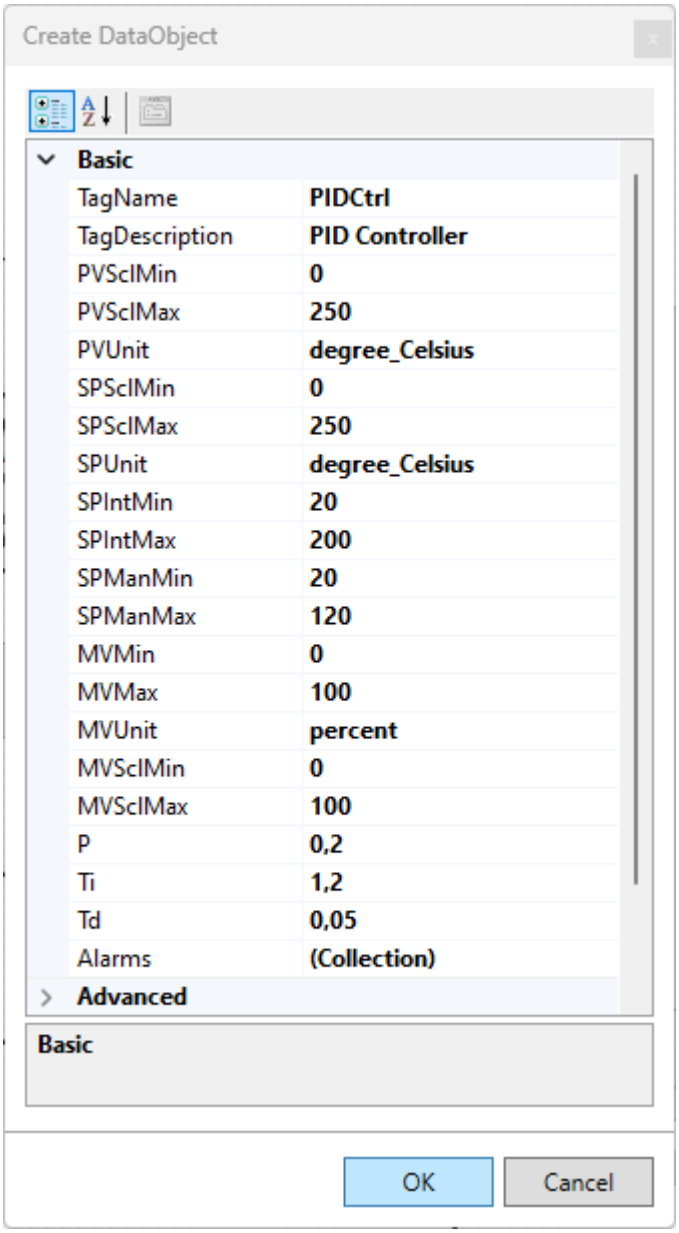
The `VisualObject` is displayed in the **Preview** area. There you also have the option of rotating the object in 90° increments using the arrows.

If you want to recreate the `DataObject`, click on **Next**. This opens the dialog for initializing the `DataObject`. The `VisualObject` is then created in the picture.

4.6 Creating new DataObjects

You can also create a new `DataObject` via the context menu of the `DataObjects` aspect.

1. Select the type in the dialog box that opens automatically. (In the picture e.g. PID controller).
2. Specify the initial values.



⇨ The result in the automatically generated PLC project (see chapter Automatic PLC code generation [▶ 20]) then looks like this:

```
{region 'ActiveElements'}
    PIDCtrl : FB_MTP_PIDCtrl := (
            TagName := 'PIDCtrl',
            TagDescription := 'PID Controller',
            MV := 0,
            MVMan := 0,
            MVMax := 100,
            MVMin := 0,
            MVSclMax := 100,
            MVSclMin := 0,
            MVUnit := E_MTP_Unit.percent,
            OSLevel := 0,
            P := 0.2,
            PV := 0,
            PVSclMax := 250,
            PVSclMin := 0,
            PVUnit := E_MTP_Unit.degree_Celsius,
            SP := 0,
            SPInt := 0,
            SPIntMax := 200,
            SPIntMin := 20,
            SPMan := 0,
            SPManMax := 120,
            SPManMin := 20,
            SPSclMax := 250,
            SPSclMin := 20,
            SPUnit := E_MTP_Unit.degree_Celsius,
            SrcChannel := FALSE,
            SrcIntAct := FALSE,
            SrcIntAut := FALSE,
            SrcIntOp := FALSE,
            SrcManAct := FALSE,
            SrcManAut := FALSE,
            SrcManOp := FALSE,
            StateAutAct := FALSE,
            StateAutAut := FALSE,
            StateAutOp := FALSE,
            StateChannel := FALSE,
            StateOffAct := FALSE,
            StateOffAut := FALSE,
            StateOffOp := FALSE,
            StateOpAct := FALSE,
            StateOpAut := FALSE,
            StateOpOp := FALSE,
            Td := 0.05,
            Ti := 1.2,
            WQC := 255);
{endregion}
```

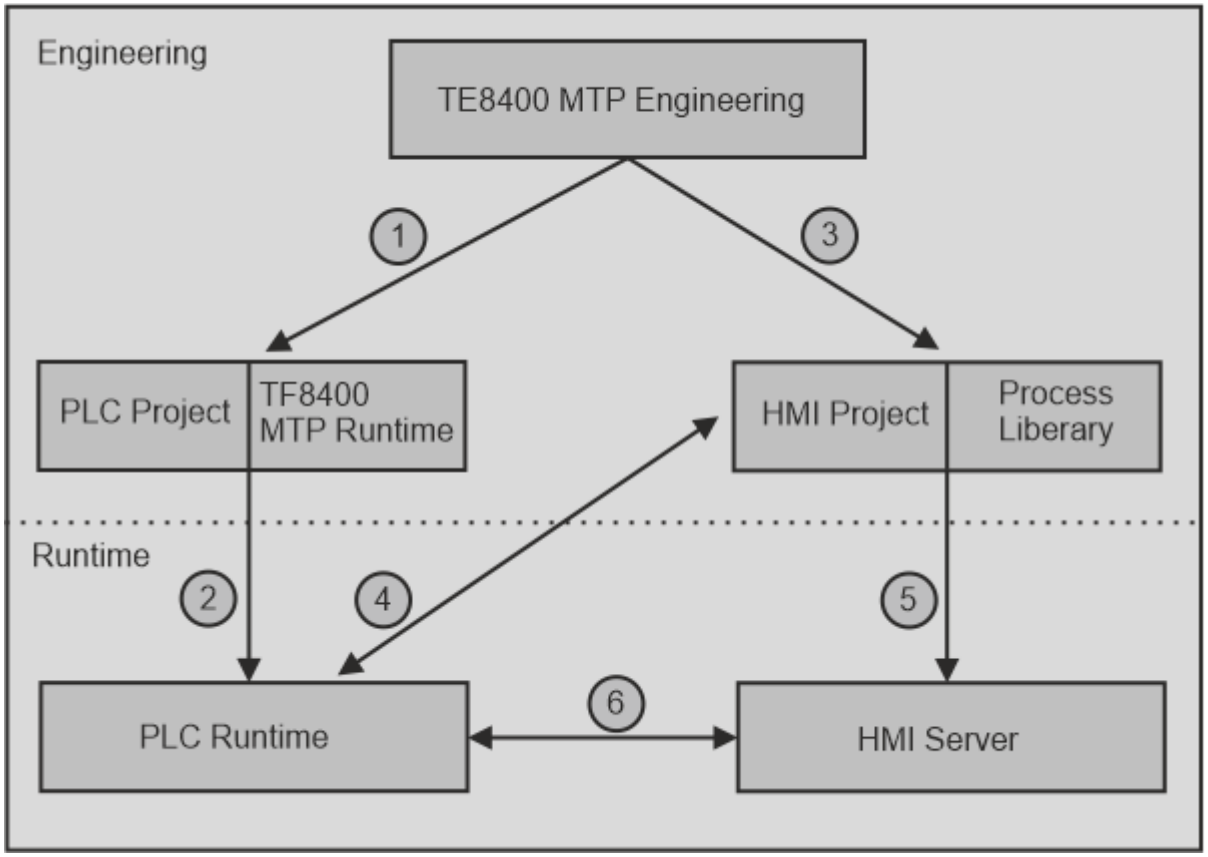
4.7 MTP export

The MTP file is created via the context menu of the project with `Export MTP`. Exporting is possible at any time without the need for a PLC or HMI project. However, in order for the module to be successfully integrated into a POL, you must first create and activate the PLC project and initialize the OPC UA server.

4.8 Generating PLC and HMI projects

You can use MTP Engineering to generate a PLC project and an HMI project from the objects in the module definition.

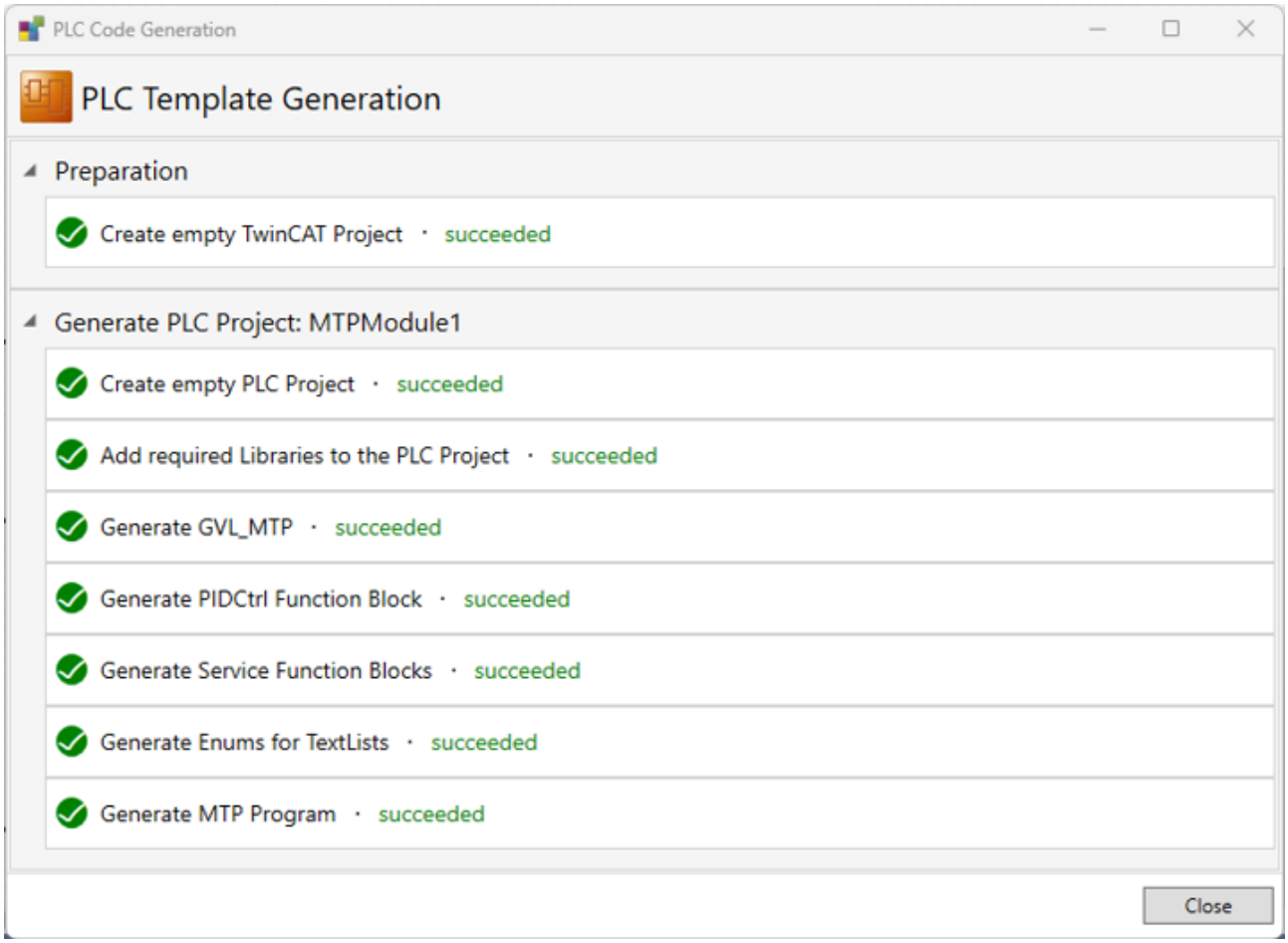
You can integrate these projects into an existing project or generate a new project. You can select the existing projects in the module properties under **PLC Reference** or **HMI Reference**.



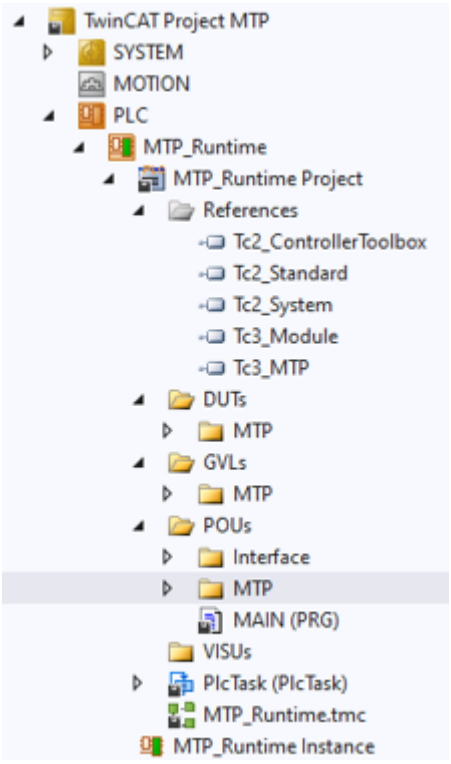
1. Generate/integrate the PLC template (using the TF8400 MTP Runtime).
2. Activate the PLC project.
3. Generate/integrate the HMI project. (using the `TwinCAT HMI Process Library`)
   ⇨ By activating the PLC project beforehand, the HMI controls are automatically mapped to the function blocks in the PLC.
4. Use the Live View connection to test the project
5. Transfer the HMI project.
   ⇨ The Live View connection for operating the project has been created
6. Use the runtime connection to operate the project.

4.9 Automatic PLC code generation

Create a new PLC project that serves as a basic framework for programming by selecting **Generate PLC Template** in the project context menu. Declare, instantiate and initialize all `DataObjects`, `Services`, `ProcessValues` and `TextLists`.



The automatically generated program code is located in folders with the name **MTP**. The folder structure under TwinCAT PLC is as follows:



Structure:

- References
  - All installed libraries
- DUTs → MTP
  - All text lists: Enum definition, service interactions and service positions
- GVL → MTP
  - Instances of the `DataObjects` and services
- POU → MTP → MTP.PRG
  - Calling `DataObjects` and services
- POU → MTP → Folder with service name
  - Declaration of the service and the procedures

The automatically generated code contains comments with notes and warnings that describe which areas will be overwritten during the next code generation and where changes can/should be made:

```
1  // ****************************************************************************
2  // * This file was automatically generated by TwinCAT 3 MTP Engineering.      *
3  // *                                                                          *
4  // * WARNING: Do not edit this function block manually. Any changes made      *
5  // * will be overwritten the next time the code generation process runs.      *
6  // *                                                                          *
7  // * INFO: Implement service states in methods. Variables related to          *
8  // * this procedure should be added here. This function block can extend      *
9  // * existing procedure function blocks.                                      *
10 // ****************************************************************************
11 FUNCTION_BLOCK FB_MTP_Filling
12 VAR_GENERIC CONSTANT
13     cProcedures_Filling : UINT := 1;
14     cConfParameters_Filling : UINT := 1;
```

```
1   (* Link inputs and outputs as needed.
2    * Future code generation will not change existing code.
3    * Code for new function blocks will be appended at the end. *)
4
5   {warning 'Add linking for P01'}
6   {region P01}
7       // Input Pump Source 1(Analog Drive)123456789
8       P01(
9           RpmFbk := P01.Rpm,
10          FwdFbk := P01.FwdCtrl AND P01.RpmFbk > 0,
11          RevFbk := P01.RevCtrl AND P01.RpmFbk > 0
12      );
13  {endregion}
14
15  {warning 'Add linking for P02'}
16  {region P02}
17      // Input Pump Source 2(Analog Drive)
18      P02(
19          RpmFbk := P02.Rpm,
20          FwdFbk := P02.FwdCtrl AND P02.RpmFbk > 0,
21          RevFbk := P02.RevCtrl AND P02.RpmFbk > 0
22      );
23  {endregion}
24
25  {warning 'Add linking for P03'}
26  {region P03}
27      // Output Pump (Analog Drive)
```

Error List

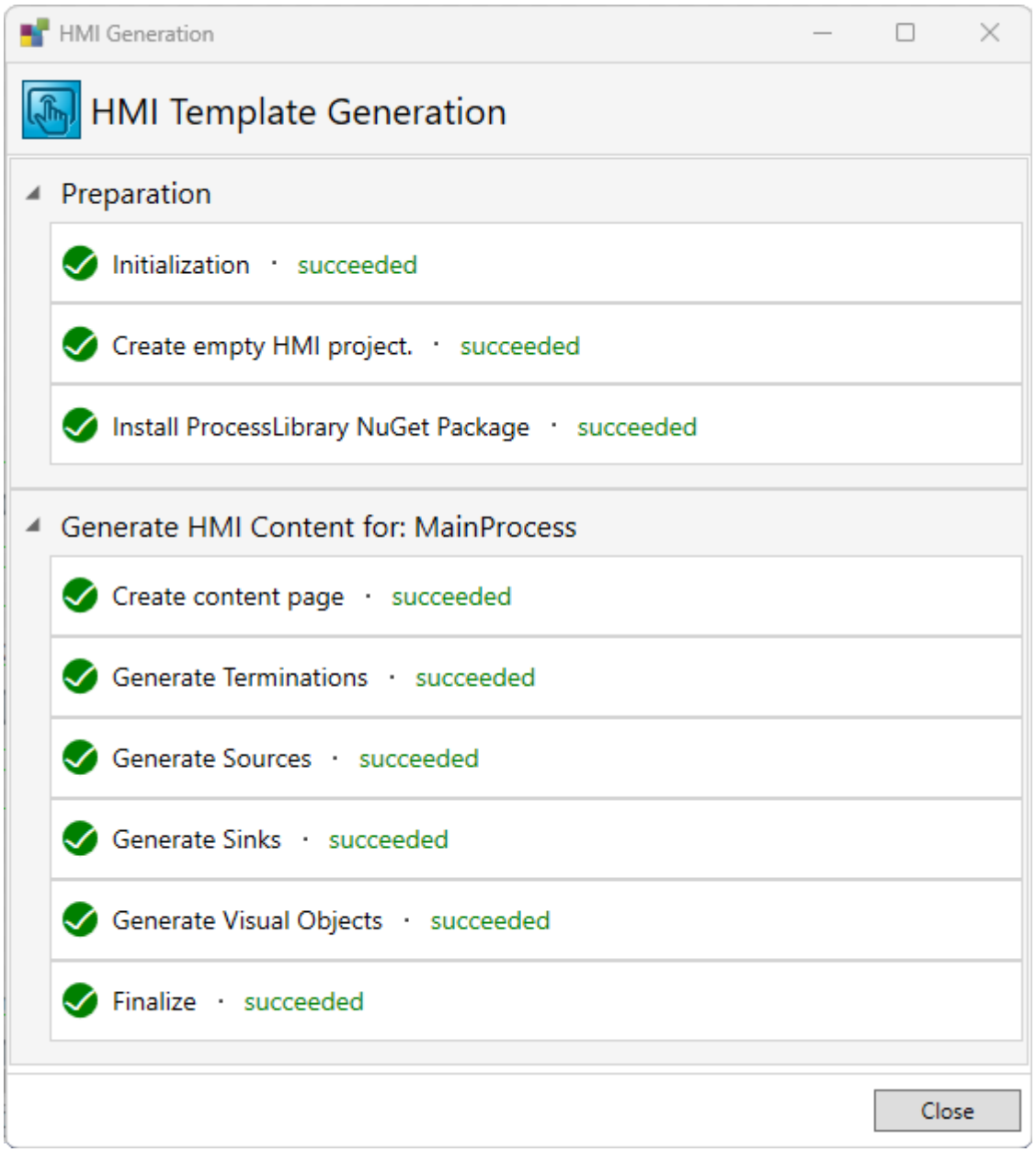
Entire Solution | 0 Errors | 36 Warnings | 0 of 97 Messages | Clear | Build + IntelliSense

Description
C0373: add state implementation
C0373: add state implementation
C0373: If your strings contain special characters, such as ß, ä, ü, ö or any other non-standard characters, you need to enable
C0373: Add linking for AlarmMgmt_PeaInformationLabel_PowerSupply
C0373: Add linking for AlarmMgmt_Filling_Holding
C0373: Add linking for AlarmMgmt_Filling_Stopping
C0373: Add linking for AlarmMgmt_Filling_Aborting
C0373: Add linking for P01
C0373: Add linking for P02
C0373: Add linking for P03
C0373: Add linking for T01

TwinCAT HMI Configuration | Error List | Output

4.10 HMI generation

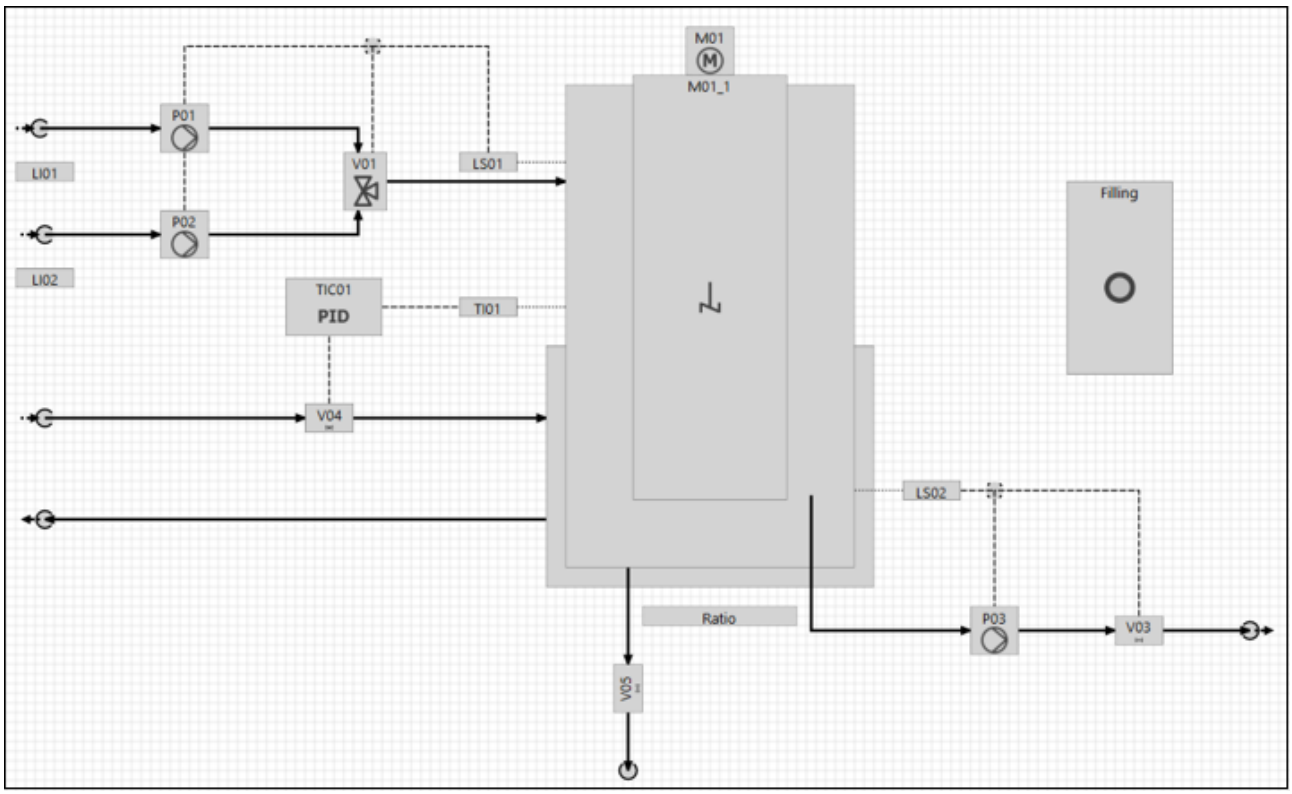
Using the project context menu **Generate HMI Template**, you can create a new HMI project with the corresponding content or generate the content in an existing HMI project.

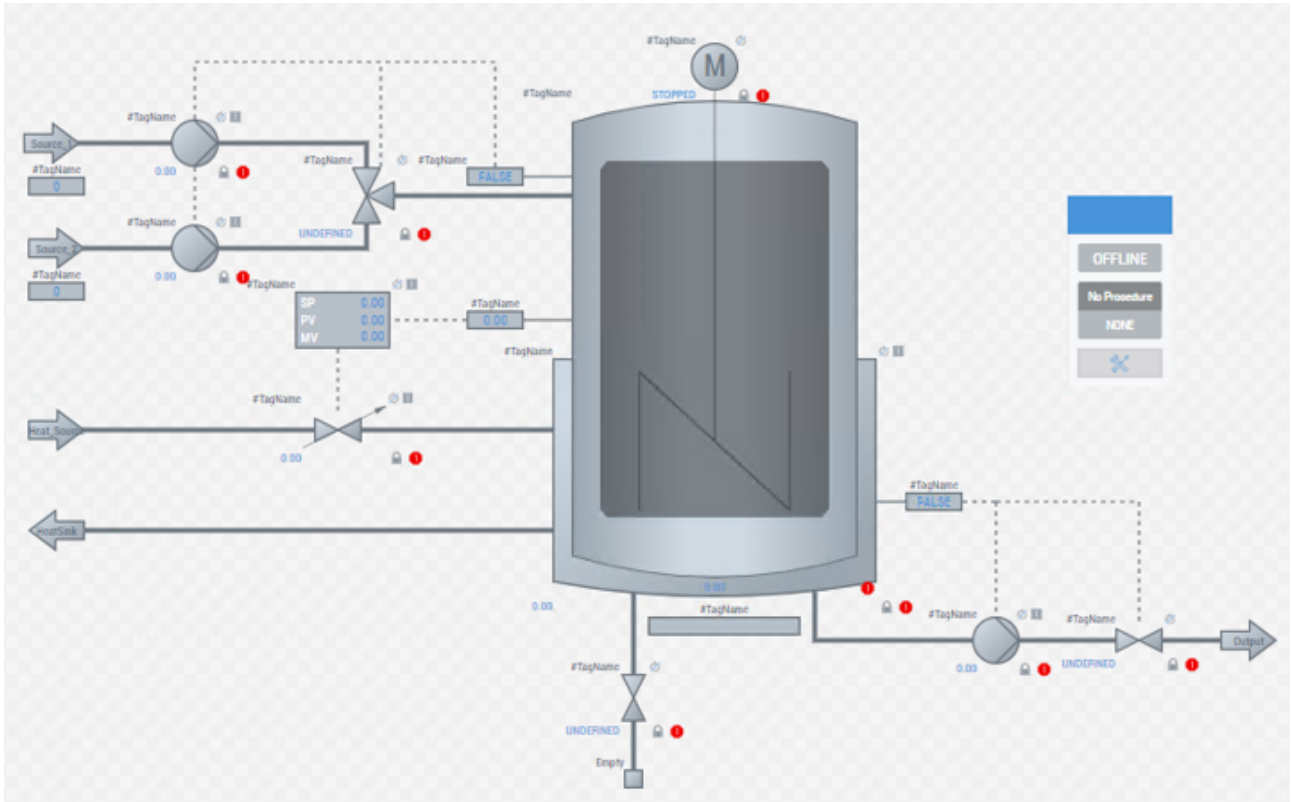


Generate and activate the PLC project before HMI generation so that the automatic mapping between PLC function blocks and HMI controls is implemented,

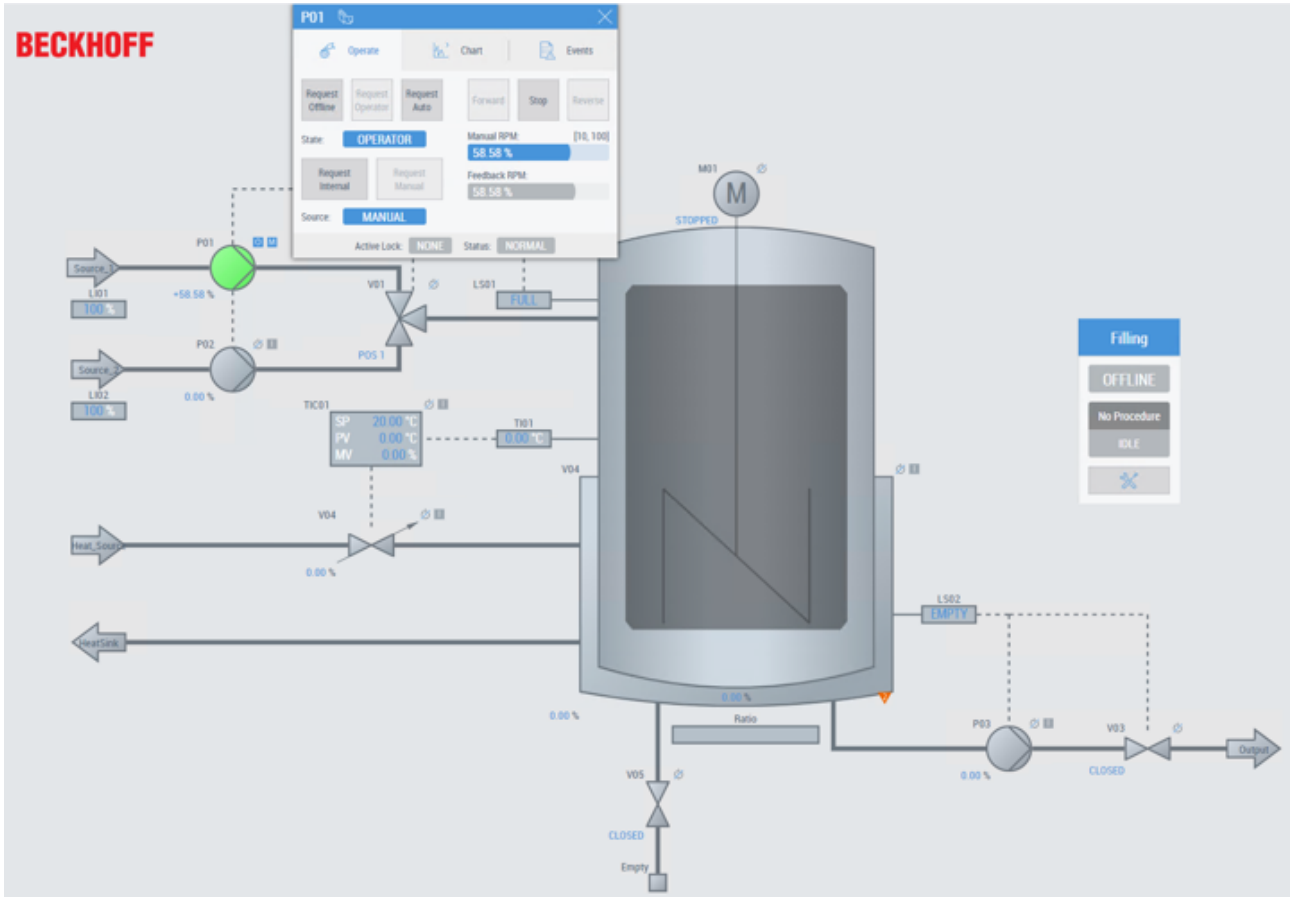
The `Region` of the `Desktop.view` in the new HMI project is created automatically and linked to the newly created content. The new page is then opened automatically to create the `Historize Settings` in the background.

Below you can see the HMI picture in MTP Engineering and the TwinCAT HMI page created from it:





You can use Live View to test the operation in TwinCAT HMI Engineering or you can transfer the HMI project to a TwinCAT HMI server:


BECKHOFF
P01
Operate
Chart
Events
Request Offline
Request Operator
Request Auto
Forward
Stop
Reverse
State:
OPERATOR
Manual RPM:
[10, 100]
58.58 %
Request Internal
Request Manual
Feedback RPM:
58.58 %
Source:
MANUAL
Active Lock:
NONE
Status:
NORMAL
M01
STOPPED
P01
Source_1
LI01
100 %
+58.58 %
V01
LS01
FULL
P02
Source_2
LI02
100 %
0.00 %
POS 1
TIC01
SP 20.00 °C
PV 0.00 °C
MV 0.00 %
TI01
0.00 °C
V04
Heat_Sour
0.00 %
HeatSink
Filling
OFFLINE
No Procedure
IDLE
LS02
EMPTY
0.00 %
0.00 %
Ratio
P03
0.00 %
V03
CLOSED
Output
V05
CLOSED
Empty

Trademark statements

More Information:
www.beckhoff.com/te8400

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