BECKHOFF New Automation Technology

Manual | EN

TE3850

TwinCAT 3 | Machine Learning Creator

2026-07-06 | Version: 1.0.0

Table of contents

1 Foreword

1.1 Notes on the documentation

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

1.4 Documentation issue status

Version	Changes
1.0.0	First release

2 Overview

The TwinCAT **M**achine **L**earning **C**reator (MLC) is a web-based engineering environment for creating AI models for industrial applications. The software guides the user through the standardized workflow of data preparation, model training, validation, and deployment.

The goal is to automatically generate AI models without requiring in-depth knowledge of data science or machine learning. The focus is on supporting automation and process experts in implementing data-driven applications.

Typical applications include AI-based quality inspection, classification, and process monitoring using image data.

You can find more current application examples on the MLC start page at: www.beckhoff.com/mlc

3 Requirements

Web browser

A current web browser is required to use the TwinCAT Machine Learning Creator. The latest versions of popular browsers such as Google™ Chrome™, Microsoft Edge, and Mozilla Firefox are supported.

MyBeckhoff User Account

A valid MyBeckhoff user account is required to authenticate in the TwinCAT Machine Learning Creator.

If you haven't created a user account yet, you can do so here:
https://www.beckhoff.com/en-en/mybeckhoff-registration/index.aspx

The MyBeckhoff user account is a centralized user account for a range of Beckhoff services. For example, you can also use it to download TwinCAT products via the TwinCAT Package Manager or to log in to MX-Designer.

4 Licensing

Overview

The TwinCAT Machine Learning Creator is provided as a subscription-based cloud service. Use is governed by user-specific licenses with a defined term and a limited allocation of computing resources (compute hours).

For production environments, various licensing options are available, which differ in terms (monthly or annual), available compute hours, and scope of use.

The licenses allow users to create their own AI models based on image data. Depending on the license type, the generated models can either be run exclusively in TwinCAT or exported as ONNX files for external runtime environments.

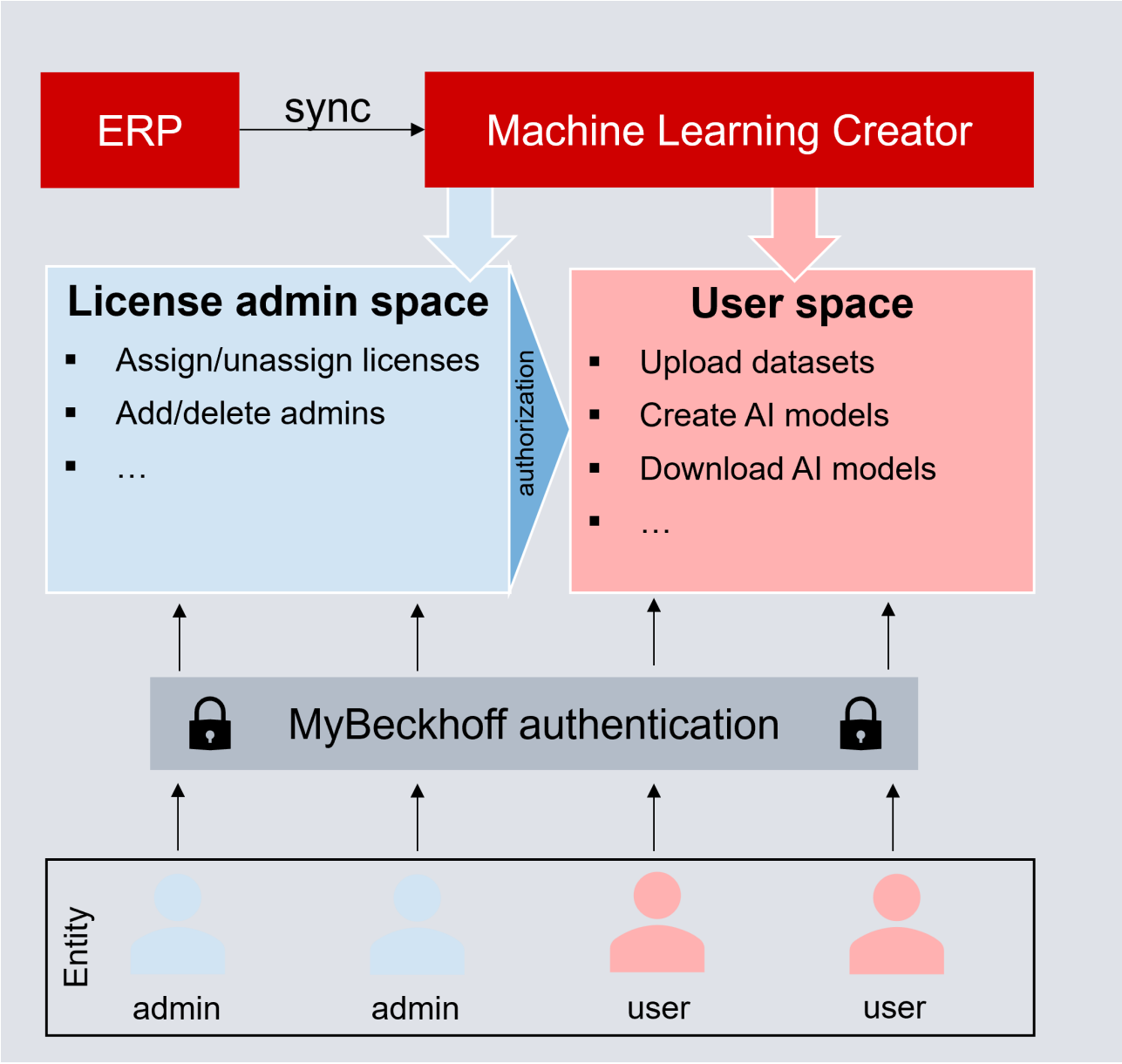
Additional computing resources can be provided through expansion packages. Unused compute hours expire at the end of the respective runtime.

For more information, visit the product page at: https://www.beckhoff.com/te3850

User onboarding

License management is handled centrally via Beckhoff ERP and is automatically synchronized with the TwinCAT Machine Learning Creator. Manual license transfer using license request and response files, as is common in many TwinCAT products, is not required.

Access to the platform is granted via a MyBeckhoff user account. Licenses are managed within an organization by designated license administrators [▶ 13]. They assign available licenses to individual users [▶ 16] or revoke them as needed.



License administrators can view all licenses assigned to their organization and control access to functions based on the scope of each license.

When a customer places their very first order for an MLC license, an initial license administrator is designated. License administrators can be added and removed independently as described below.

As with all other Beckhoff products, MLC licenses are ordered through Beckhoff sales.

5 Workflows and UI views

5.1 Navigation

Navigation elements

The TwinCAT Machine Learning Creator uses two main navigation elements:
User navigation and the side menu.

User navigation

The user navigation is located in the upper-right corner of the application's header.

General user and administrative functions are available through the user navigation.
In addition, you have the choice of selecting the user interface language from the header.

The following sections can be opened via the user navigation (using "en-US" as an example here):

- Projects Overview
- Recent Updates
- My Profile
- License Admin (for license administrators only)
- Contact
- Sign Out

Side menu

The side menu is located on the left side of the user interface.

You can navigate within a project using the side menu.

The side menu can be expanded and collapsed.

Depending on the context, the following areas, among others, are available:

- Dataset [▶ 18]
- Models [▶ 22]
- Performance

You can switch between the various project views using the individual menu entries.

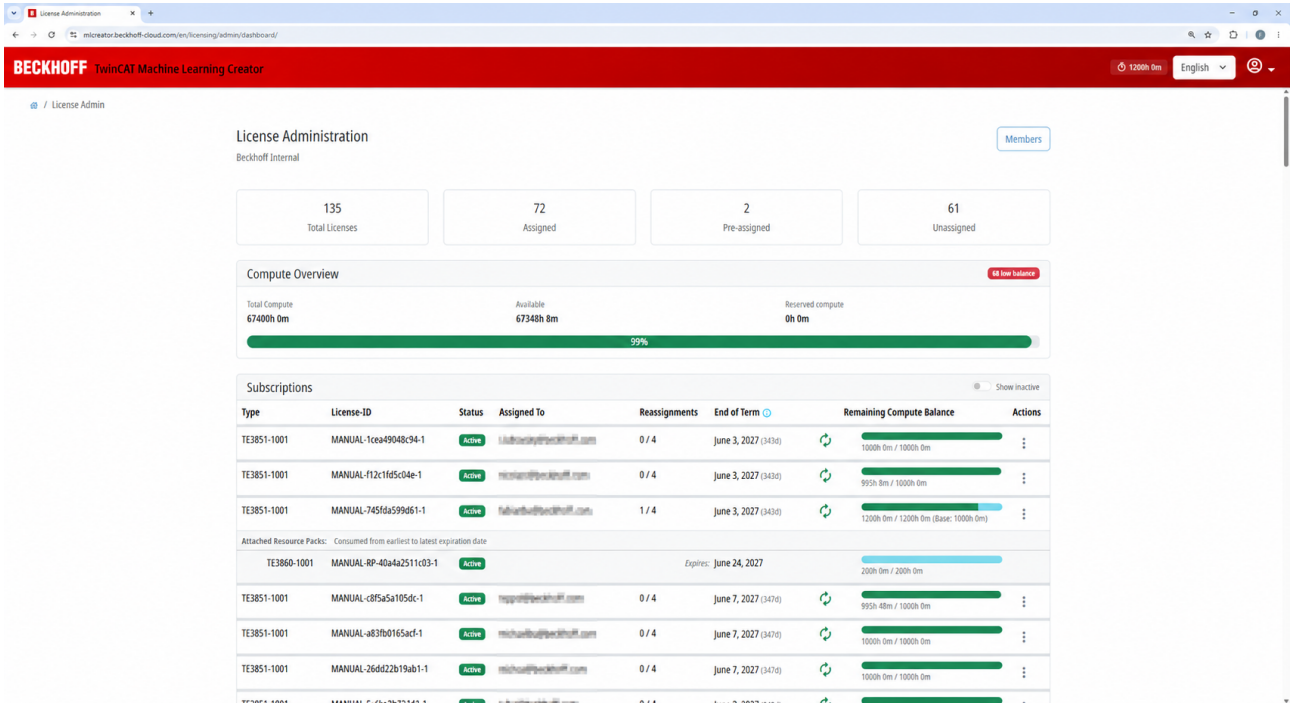
Home button (Beckhoff logo)

Clicking the Beckhoff logo opens the License Administrator Panel [▶ 13] for license administrators and the Project Overview [▶ 17] for users.

5.2 Workflows for license administrators

License administration

The License Administrator Panel is used for the centralized management of all MLC licenses and resources assigned to the enterprise. Access is restricted to users with the "License Administrator" role.



Members

Users can be assigned to the respective enterprise using the Members [▶ 15] button. Only members can be designated as license administrators or users.

Overview

The top section displays aggregated metrics on license distribution:

- **Total Licenses**: Total number of available licenses
- **Assigned**: Number of currently assigned licenses
- **Pre-assigned**: Number of licenses already assigned (member has not yet confirmed)
- **Unassigned**: Number of unassigned licenses

These figures provide a quick overview of the current license distribution.

Compute Overview

The "Compute Overview" section displays the available computing resources:

- **Total Compute**: Total compute quota for all licenses and Resource Packs
- **Available**: Remaining compute quota
- **Reserved compute**: Resources that have already been reserved but not yet used

The progress bar displays the current utilization. If necessary, a note indicating a low remaining quota will be displayed directly in this area.

Subscriptions

The "Subscriptions" table represents all license assignments in detail:

- **Type**: License type (e.g., TE3851-1000)
- **License ID**: Unique license identifier
- **State**: Actual state (e.g., Active, Grace, Expired)
- **Assigned To**: Assigned User
- **Reassignments**: Number of remaining reassignments within the runtime
- **End of Term**: Expiration date of the current term

- **Remaining Compute Balance**: Remaining compute quota for the license

Actions

Depending on your license state, the following actions [▶ 16] are available via the three-dot menu:

- **Assign:** Assign a license to a user
- **Reassign**: Transferring the license to another user
- **Unassign**: Remove the current assignment
- **Attach Pack**: Assign a Resource Pack to extend the compute quota

Attached Resource Packs

When a resource pack has been assigned to a license, it appears in the *Attached Resource Packs* section directly below the corresponding subscription. The following applies:

- The resource pack is tied to a specific license instance. It is not assigned to a specific user, but rather to the license.
- The assigned pack increases the license's available compute quota.
- The display shows the pack along with its license ID, state, and expiration date.
- The license's "Compute" display takes into account the additional quota from the pack.

Unattached Resource Packs

This section shows available resource packs that have not yet been assigned:

- **Type**: Resource pack type, e.g., TE3860-1001
- **License ID**: Unique identifier for the resource pack
- **Compute**: Included compute hours
- **Expiry**: Expiration date
- **State**: Actual state
- **Compatible Licenses**: The types of licenses with which the resource pack can be used

You can use the switch **Show inactive** to show or hide inactive resource packs. Resource packs are displayed in the "Attached Resource Packs" section only after they have been assigned to a compatible license.

5.2.1 Manage members of the organization

Click the "Member" button to go to the member management screen.

New users are invited to join the organization by a license administrator. To do this, enter the new user's email address via **Invite Member**. Optionally, a license can be pre-assigned here. The user is then automatically assigned this license as soon as they accept the invitation to join the organization.

MyBeckhoff account required

As a general rule, the email address you use should already be associated with an existing MyBeckhoff ID. If the user does not yet have a MyBeckhoff account, they must first create an account using this email address.

After the invitation is sent, the user receives a verification email. The invitation must be confirmed via the link provided.

Until confirmation, the user's status is **Pending**.

Once confirmation is successful, the state changes to **Member**. After that, MLC licenses can be assigned to the user (pre-assigned licenses are automatically assigned as soon as the user reaches Member state).

License administrators can manage member roles (under **Actions**):

- **Promote**: Promotes a member to license administrator.
- **Demote**: Removes administrator privileges and resets the user's state to *Member*.

- **Remove**: Removes the member from the organization.

A license administrator cannot delete their own administrator rights. This ensures that there is always at least one license administrator for the organization.

5.2.2 Manage licenses

Subscription licenses and Resource Packs are managed centrally through the License Administration panel.

Assign subscription licenses

Subscription licenses can only be assigned to users with **Member** state. Assignments are made by a license administrator.

Each subscription license can be assigned to only one user at a time. The current assignment is displayed in the **Assigned To** column.

You can use **Reassign** to assign a license to another user. The number of possible reassignments during the runtime is limited and is represented in the **Reassignments** column.

The number of reassignments allowed depends on the license type:

- Annual subscription licenses: up to four assignments per runtime
- Monthly subscription licenses: one assignment per runtime

Unused reassignments expire at the end of the respective subscription period.

Remove license assignment

You can use **Unassign** to remove the current user assignment for a subscription license. The license will then be available for reassignment within the permitted reassignment limits.

Assign resource packs

Resource packs expand the available compute quota for a subscription license. Resource packs are not assigned directly to users, but are always assigned to a compatible base license.

Unassigned resource packs are displayed in the **Unattached resource packs** section.

Attach pack allows you to link a resource pack to a subscription license. Once assigned, the resource pack remains tied to this base license for its entire runtime and cannot be transferred to other licenses.

Compute hours are automatically consumed based on the earliest expiration date. This ensures that the resources with the earliest expiration dates are used first.

5.3 Workflows for users

Workflow overview

The TwinCAT Machine Learning Creator provides a complete workflow for creating and deploying AI models within a web-based user interface.

The workflow consists of the following steps:

Create project

First, a project is created on the platform. The project defines the task types (e.g., image classification) and serves as a framework for data sets and models.

Create a data set

First, image data is uploaded to the platform. The images are then (or immediately upon upload) tagged with labels that define the desired target class. The data set forms the basis for subsequent model training.

Configure model training

For training, a new model is created and linked to a data set. In addition, training-related parameters, as well as target hardware, target software, and runtime requirements, can be defined.

Run a model training session

Once training starts, the platform automatically generates an AI model based on the training data provided.

Validate and analyze the model

Once training is complete, various analysis and explainability functions are available. These include, among other things, statistical evaluation methods, confidence intervals, confusion matrices, and visual representations that provide insight into the model's decision-making process.

Export model

Trained models can then be downloaded. In addition, PLCopen-XML can be exported to simplify integration into TwinCAT projects.

5.3.1 Create projects

Project overview

After logging in, the project overview is displayed. Here, you can manage existing projects and create new ones.

A project serves as an organizational unit for data sets, training runs, and AI models.

Creating a new project

Click **Create project** to create a new project. This process involves defining the project name and optional metadata.

Open project

A project is opened using **Open**. Afterward, you can create data sets, train models, and analyze results.

Edit project

You can edit the project metadata using the icon:

- Project name
- Project description

Add project to favorites

You can mark a project as a favorite by clicking the icon. Favorite projects are displayed in the **Favorites** section.

Delete project

You can delete a project by tapping the icon.

Managing tags

You can add tags to projects using the **Add tag** option. Tags are used to organize content and make it easier to filter and find projects.

Search and filter

You can search for projects by name or metadata using the search box. In addition, filter functions are available to narrow down the projects displayed.

Sort

By default, the project list is sorted chronologically by the date of the last modification. Projects with the most recent activity are displayed first.

5.3.2 Create and manage data sets

Data sets form the basis for training AI models. A data set contains the image data as well as the corresponding labels for the desired target classes.

Create a data set

Create data set creates a new data set. A data set name must be specified; additional metadata can be entered optionally.

Data set information

The left pane displays the most important information about the data set:

- Name of the data set
- Description
- Time of the last saved version
- Number of files included
- Training and test split
- Results of an initial ad hoc data set validation

By default, the data set is automatically split into training and test data:

- 80% training data
- 20% test data

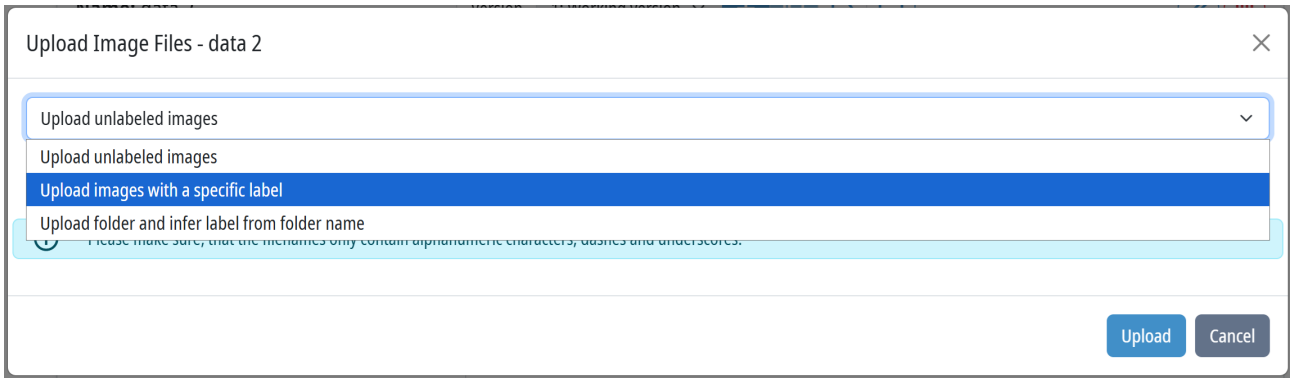
The test data is used exclusively for subsequent model validation.

Actions

Uploading images

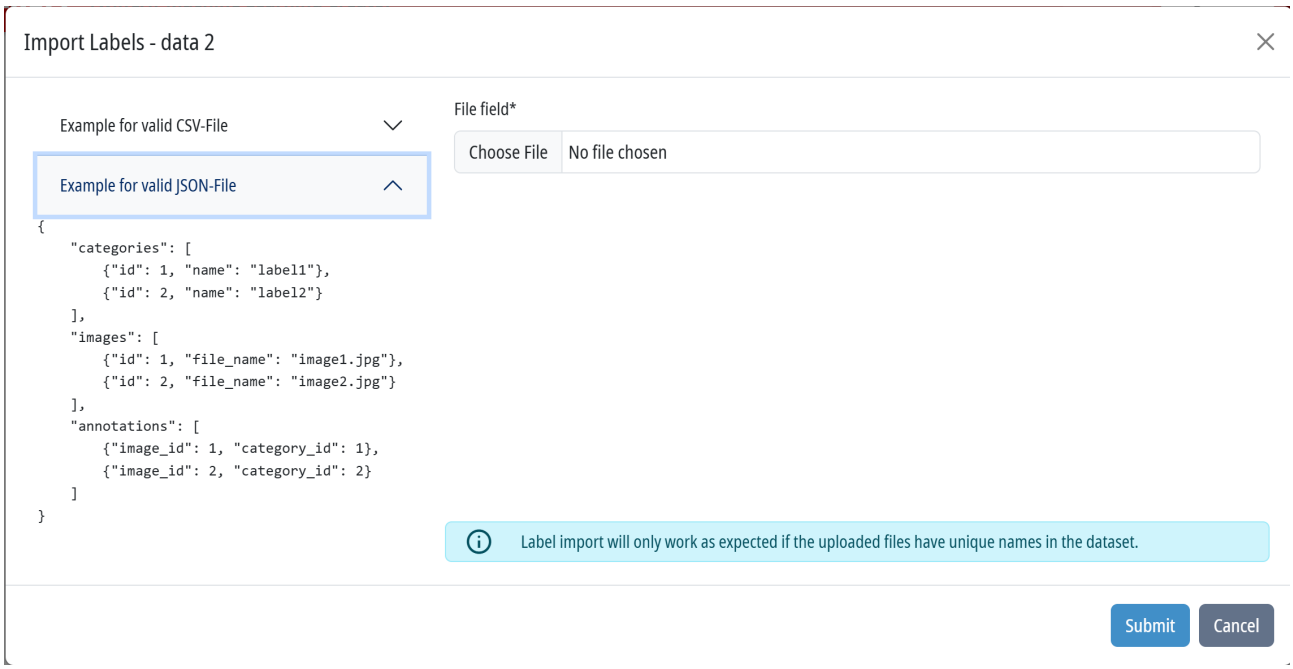
You can import image files into the data set using **Upload image files**.

Common image formats are supported.



Import labels

You can import existing label information using **Import labels**. This allows existing annotations from external tools to be imported.



Alternatively, data can also be labeled on the platform [▶ 19].

Data set versions

Display and choice of the data set version. Data set versions represent a fixed and reproducible state of the data set and serve as the basis for model training.

A version that has not yet been versioned is displayed as a “working version”.

Create a new data set version

Click the **+ icon** to create a new version of the data set.

This saves the current state of the data set. Only saved data set versions can be used for AI model training.

Clone a data set

You can duplicate a data set by clicking the **icon**.

The cloned data set contains all the images, labels, and metadata from the original data set and can be further processed independently.

Label manager

Click the icon to open the Label Manager. Here, you can create, edit, and manage class labels.

File explorer

Click the icon to open File Explorer.

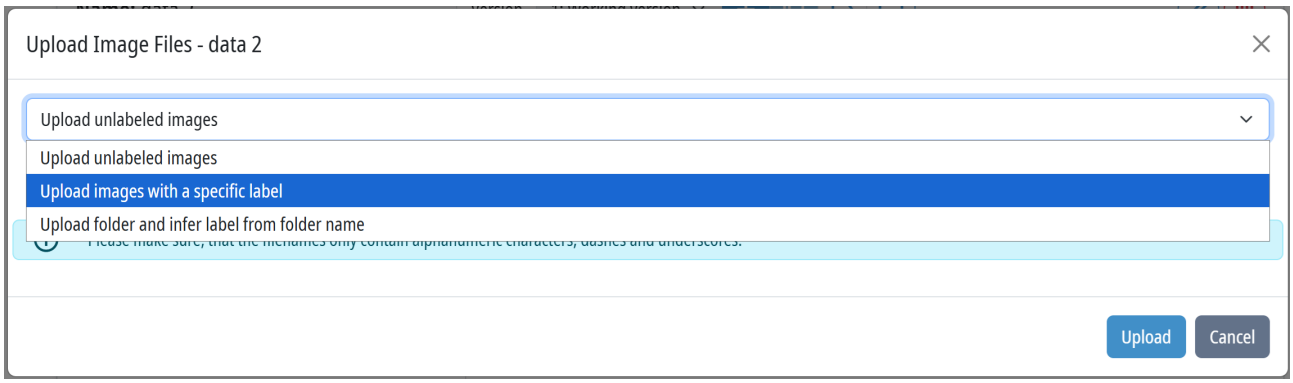
File Explorer allows you to view and navigate all files within the data set. This allows images to be managed, filtered, and reviewed individually. In addition, you can create and edit labels in File Explorer.

5.3.2.1 Labeling options for image classification

For image classification data sets, there are various options available for creating and managing labels.

Labeling during file upload

Labels can be assigned while the image data is being uploaded.



Label assignment for an upload batch

When uploading, you can specify that all selected images be assigned to a specific label.

By uploading data in multiple batches with different labels, you can create a data set with multiple classes.

Example:

- Upload Batch 1 → Label OK
- Upload Batch 2 → Label NOK

Labeling via folder structure

Alternatively, images can already be organized locally into subfolders.

When uploading a directory tree, the name of each subfolder is automatically used as a label.

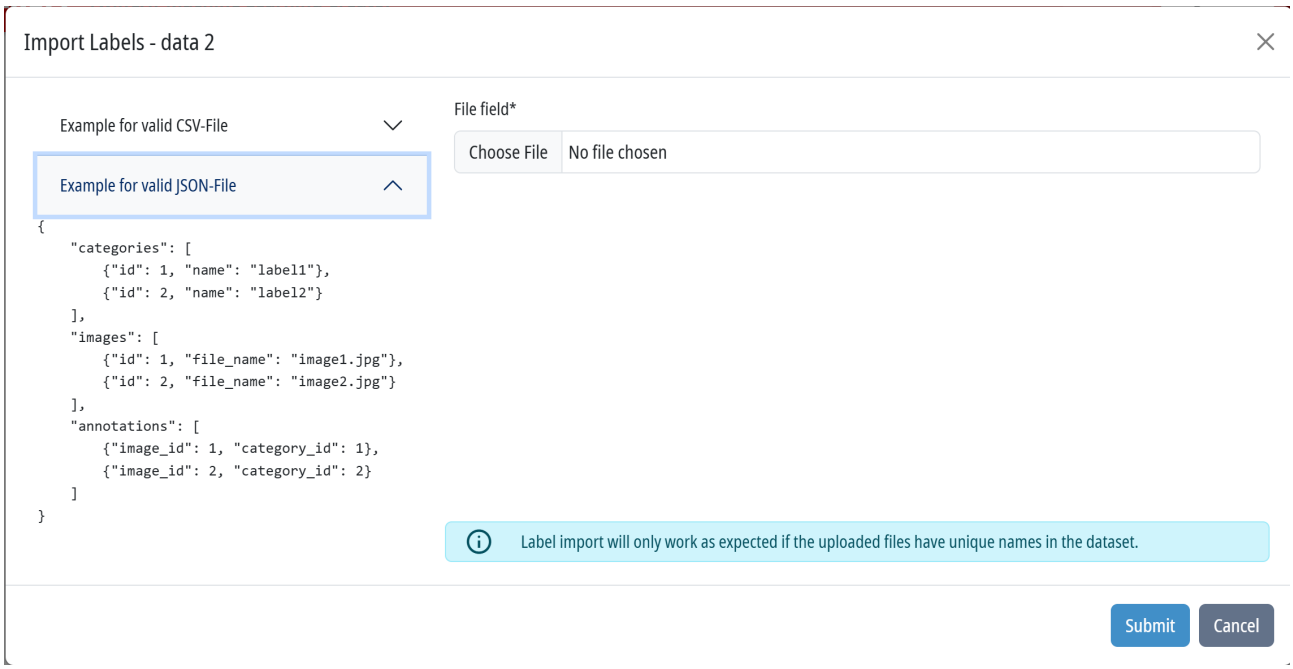
Example:

```
dataset/
├── OK/
│   ├── img_001.png
│   └── img_002.png
├── NOK/
│   ├── img_003.png
│   └── img_004.png
```

In this example, the labels "OK" and "NOK" are automatically generated and assigned to the respective images.

Importing external labels

Labels can be imported from external annotation tools.



The import is done using **Import labels.**

Supported formats:

- CSV
- COCO

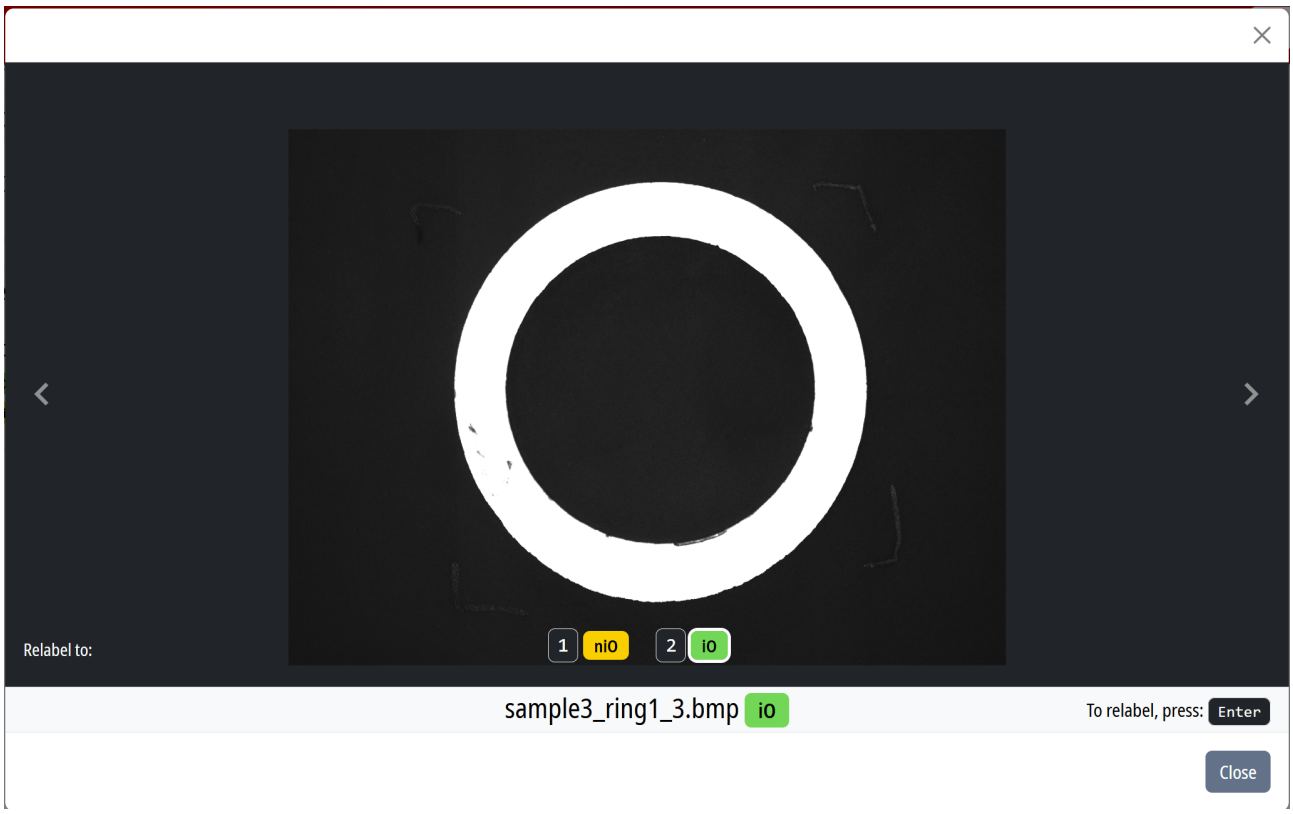
This allows existing annotations from external workflows to be imported.

Labeling in TwinCAT Machine Learning Creator

Labels can be created and edited directly within the TwinCAT Machine Learning Creator.

Options available through the **Label manager**:

- Create new labels
- Rename labels
- Manage existing labels



In **File explorer**, you can label images individually. To do this, make a choice of an image and then press the **Enter** key to switch to label mode. In label mode, images can be assigned to existing classes or relabeled. You can use the left and right arrow keys to quickly navigate through the data set.

5.3.3 Model training

Create a model

The **Create model** option is used to create a new AI model and configure it for training.

As a requirement to create a model, you must have already uploaded a data set and saved at least one version of that data set. Only versioned data sets can be used for training.

General model information

- **Name:** Unique name of the model.
- **Description:** Optional description of the model and its intended use.

Data set configuration

- **Data set:** Choice of the data set to be used for training. Only data sets for which a version has already been created are displayed.
- **Data set version:** Choice of the data set version to use. This version represents a fixed and reproducible state of the data set. Changes made to the data set after revision control do not affect existing training configurations.

Runtime and target system configuration

- **Latency threshold in ms:** Defines the maximum allowed execution time for the AI model on the target system [▶ 24]. This value is taken into account during model creation in order to make a choice of a model that is suitable in terms of accuracy and runtime performance.
- **Target device:** Choice of the target hardware on which the model will later be executed.
- **CPU:** Choice of the CPU configuration used by the target system.
- **GPU option:** Choice of supported GPU configurations.

- **Target software:** Defines the TwinCAT target environment for subsequent inference, for example:
  - TF3820 | TwinCAT 3 Machine Learning Server
  - TF7810 | TwinCAT 3 Vision Neural Network
- **Execution provider (EP):** Defines the execution environment for the Machine Learning Server inference engine, such as CPU or CUDA® (NVIDIA® GPU).
- **Number of cores used for inference:** Specifies the number of CPU cores that will be used for subsequent model execution. When using TF7810, this value refers to isolated CPU cores within the Vision Job Pool. When using TF3820 with Execution Provider = CPU, the value corresponds to the number of available CPU cores in the user mode process of the TwinCAT Machine Learning Server. Only user mode cores are used in this process. On systems with a hybrid architecture, only performance cores (P-cores) are taken into account.

Create a model

- Click **Submit** to save the training configuration. Training must then be started on the model card.
- Clicking **Cancel** closes the dialog box without making any changes.

Model card

The model card displays the current state as well as the most important configuration and performance data for an AI model.

The card's title matches the model name.

Model actions

Various actions are available in the upper section:

- **Open analysis:** Opens the model's analysis and explainability view.
- **Download model:** Downloads the trained model.
- **Edit model:** Opens the model configuration to edit the metadata
- **Delete model:** Removes the model from the project.
- **Start training:** Starts a training run for the model. Please note the rules regarding compute hours billing [▶ 24].

Model information

- **State:** Displays the actual state of the model. Possible states include, for example:
  - Submitted
  - Waiting
  - Running
  - Finished (including a timestamp indicating when the model reached this state)
  - Error
- **Description:** Optional description of the model
- **Data set:** Displays the data set used, including the data set version. You can open the corresponding data set by clicking the search icon.
- **Labels:** Displays the classes or target categories contained in the data set.
- **Target device:** Displays the configured target hardware, including the CPU, GPU, and execution provider configurations.
- **Target software:** Displays the configured TwinCAT target environment for later inference.
- **Target latency threshold:** Displays the configured maximum target latency for model execution.

Training and performance information

- **Performance:** Displays the achieved model quality as a percentage. The metric shown is based on the test results calculated during validation.
- **Training time:** Total duration of the training process.
- **Estimated latency:** Estimated runtime of the model on the configured target hardware.

- **Training history:** The "Loss History" feature allows you to visualize the progress of the training process. This represents the training metrics for individual epochs.

5.3.3.1 Billing for compute hours

Compute hours represent the available computing resources for training AI models.

The available compute quota depends on the license type used and any optionally assigned Resource Packs. At the start of a new subscription period, the compute quota included in the license is replenished. Unused compute hours expire at the end of the respective runtime. Additional compute hours can be added using resource packs.

If the availability of compute quota is no longer sufficient, no further training sessions can be started. In this case, you must either wait until the next subscription period or assign an additional resource pack.

You can find the current value of your compute quota in the top-right corner of the header; it is displayed in hours and minutes.

Reserving compute hours

When a model training session starts, the estimated training time is first calculated based on the training configuration and the data set used.

If the estimated training time is less than or equal to the available compute quota, the training is started. In this process, the estimated compute time is initially reserved. This mechanism is similar to temporarily setting aside a budget, such as when making a credit card payment.

Once the training session is complete, you will be billed only for the actual training time used. Unused reserved compute hours are automatically released.

Insufficient compute quota

If the estimated training time exceeds the currently available compute quota, the training cannot be started.

In this case, additional compute quota must be allocated.

Billing date

A training session is always billed based on the compute quota that was available at the time the training started. The reserved compute hours remain tied to the original subscription period for the entire duration of the training.

Example:

If training is started on the last day of a subscription period and does not end until the following subscription period, the entire training duration will still be charged to the original subscription term. The reason for this is that the compute quota was already reserved when training started.

5.3.3.2 Latency threshold

The **Latency threshold** parameter can be used to define the maximum allowable inference latency of the AI model.

This allows the model training to be specifically tailored to the requirements of the subsequent machine or process runtime.

The user specifies the target hardware, the target software, and the maximum allowed runtime for model execution. The TwinCAT Machine Learning Creator takes these constraints into account during model creation and optimizes the model accordingly in terms of runtime and model complexity.

This makes it possible to create AI models that meet both the desired model quality and the required time constraints.

Example:

During a quality control check, a product is recorded by a camera and then sorted out via a rejection station. There are only 10 ms available between image capture and reaching the ejection position. In this case, the inference—including preprocessing and postprocessing—must be completed within this timeframe.

The configured latency threshold allows the AI model to be adapted to this process requirement.

Balance between model quality and latency

The configured latency threshold influences the choice and optimization of the model architecture.

If no latency threshold is specified, there is no restriction on the subsequent inference runtime. This allows the optimization process to make a choice of more complex model architectures, which generally leads to better model metrics, such as accuracy, precision, or recall.

In the engineering process, it is therefore recommended to train initially without a latency threshold. This allows the data set to be iteratively improved and evaluated until a model that is convincing from a technical standpoint is achieved.

Only then should the latency threshold be reduced gradually. This allows us to verify whether sufficient model quality can still be achieved with lower model complexity and reduced inference latency.

This approach allows for a careful balance between model performance and the real-time capability of the eventual application.

5.3.4 Validate the model

Explainability view

The Explainability view is used to analyze and validate a trained AI model.

The results shown are based on the test data split of the data set used. By default, 20% of the data set is automatically reserved as test data. The remaining 80% is used for model training.

The test data is not used during training and is used solely to evaluate the model's ability to generalize.

Model statistics

The top left section represents the model's key performance metrics.

- **Test accuracy:** Describes the proportion of all correctly classified test data. Accuracy provides a general overview of the model's overall performance.
- **Test precision:** Describes the accuracy of positive predictions. Precision measures how many samples classified as positive are actually correct.
- **Test recall:** Describes the model's ability to correctly identify relevant samples. A high recall means that only a few relevant samples are missed.
- **Test F1:** The F1 score combines precision and recall into a single metric. This metric is particularly helpful when dealing with unbalanced data sets.

Confidence

The **Confidence** section displays statistical measures of model confidence:

- Average value (Mean)
- Minimum value (Min)
- Maximum value (Max)

Confidence describes the model's certainty in making a prediction.

The Confidence Distribution visualizes the distribution of all prediction confidences as a histogram. Each bar represents a confidence interval (bin) and shows the number of predictions within that interval. This makes it possible to assess whether the model makes mostly safe or unsafe decisions.

Interactive filtering

By clicking on a histogram range, you can filter the results to a specific confidence interval. The filtered results are then displayed in the **Prediction behavior** section. Only samples within the selected confidence range will appear there. The active filter is displayed in the upper section of Prediction Behavior and can also be deleted there.

Confusion matrix

The confusion matrix shows the distribution of correct and incorrect predictions by class.

- Rows correspond to the actual classes (true labels, as labeled by humans).
- The columns correspond to the predicted labels.

The diagonal of the matrix contains correctly classified samples. Values outside the diagonal represent misclassifications.

The size of the matrix depends on the number of classes in the data set.

Interactive filtering

By clicking on individual matrix elements, you can filter for specific combinations of predictions. This makes it possible to analyze misclassified samples or correctly classified samples in a targeted manner. The filtered results are represented in the **Prediction behavior** section.

Prediction behavior

The "Prediction Behavior" area displays all samples from the test data set, including the prediction results, model confidence, and attention map.

Correct and incorrect predictions can be shown or hidden separately. In addition, the results can be sorted by confidence level.

Attention maps

For each sample displayed, an attention map can optionally be represented. The visualization highlights the image areas that contributed most significantly to the AI model's decision. This makes it clear which characteristics were relevant for the classification. You can disable the Attention Map using the switch in each image.

5.3.5 Model download

Trained AI models, including integration artifacts, can be exported via the model download feature.

Choose software target

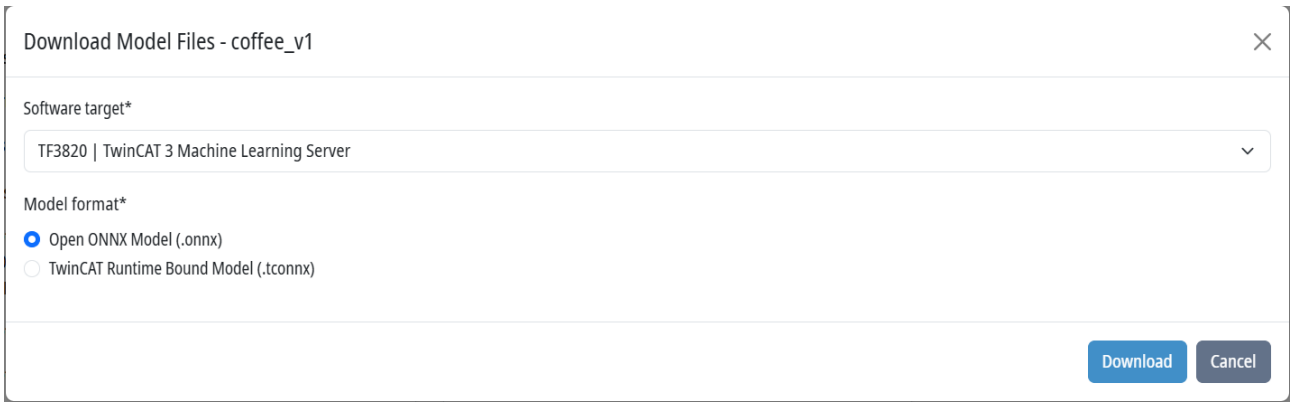
In the **Software target** field, you make a choice of the target software for generating the PLC code.

By default, the target software that was defined during model creation is used.

The target software can be changed again during the download. In this case, the PLC code is generated for the newly selected runtime environment. It is important to note that the latency estimate calculated during training is valid only for the target software that was originally configured.

Choose model format

Depending on the type of license used, different model formats are available.



Open ONNX model (.onnx)

The model is exported as a standardized ONNX file.

ONNX is an open, framework-independent format and can be used in any compatible runtime environment or AI framework.

TwinCAT runtime bound model (.tconnx)

The model is exported as a TwinCAT-linked tcONNX model.

This format is intended exclusively for use within TwinCAT 3.

Contents of the download package

The download is a ZIP archive containing several files for integrating and documenting the AI model.

AI model

Contains the trained model in either ONNX or tcONNX format, depending on the selected export format.

model_card.json

- Describes the AI model that was created, including:
- data set splits used
- Input and output formats
- Class labels
- Timestamp
- Training and runtime information
- Latency estimates

PLCopen-XML

Includes complete PLC code for integration into TwinCAT 3.

The generated code includes:

- Image acquisition
- Image pre-processing
- AI inference
- Post-processing of the results

The file can be imported directly into TwinCAT 3.

Modelname.json

Depending on the selected software target, a specific JSON file is also generated, which is required by the inference software.

readme.md

Contains a detailed description of how to integrate the generated PLCopen XML into a TwinCAT 3 project. In addition, the necessary configuration steps and project-specific adjustments are described.

6 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®, 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.

Third-party trademark statements

Chrome, Chromium and Google are trademarks of Google LLC.

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

Mozilla and Firefox are trademarks of the Mozilla Foundation in the U.S. and other countries.

NVIDIA, NVIDIA RTX and CUDA are trademarks of NVIDIA Corporation.

More Information:
www.beckhoff.com/te3850

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