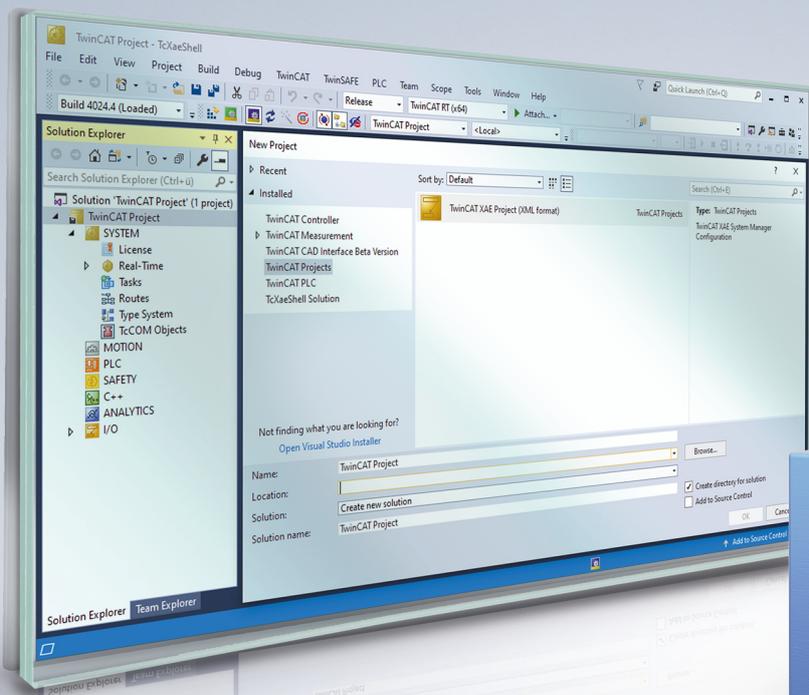


# BECKHOFF New Automation Technology

Handbuch | DE

## TF355x

TwinCAT 3 | Analytics Runtime





# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort</b>	<b>5</b>
1.1	Hinweise zur Dokumentation	5
1.2	Zu Ihrer Sicherheit	6
1.3	Hinweise zur Informationssicherheit	7
<b>2</b>	<b>Übersicht</b>	<b>8</b>
<b>3</b>	<b>Installation</b>	<b>9</b>
3.1	Systemanforderungen	9
3.2	Installation	9
3.3	Lizenzierung	12
<b>4</b>	<b>Analytics Workflow – Erste Schritte</b>	<b>15</b>
4.1	Aufzeichnung von Maschinendaten	15
4.2	Kommunikation	18
4.3	Historisierung von Daten	19
4.4	Analyse der Daten	27
4.5	24-stündige Anwendung von Analytics	32
<b>5</b>	<b>Technische Einführung</b>	<b>42</b>
<b>6</b>	<b>Konfiguration</b>	<b>44</b>
6.1	Runtime-Deployment	44
6.1.1	Algorithmeigenschaften	50
6.1.2	SPS-Code	52
6.2	HMI One-Click Dashboard	110
6.2.1	Dashboard Aufbau und Inhalte im Analytics Projekt verwalten	113
6.2.2	Angepasste und eigene Controls verwenden	123
6.2.3	Benutzerverwaltung und Zugriffsrechte konfigurieren	142
6.2.4	Dashboard-Konfiguration	145
6.2.5	Ändern eines generierten Dashboards im HMI-Engineering	152
6.2.6	Konfiguration des Dashboards zur Laufzeit im Webbrowser	152
6.2.7	Umschaltung mehrerer Maschinen im HMI Dashboard	159
6.2.8	Integration einer Sprachumschaltung	166
6.3	Analytics Reporting	169
<b>7</b>	<b>Anhang</b>	<b>170</b>
7.1	FAQ – Häufig gestellte Fragen und Antworten	170



# 1 Vorwort

## 1.1 Hinweise zur Dokumentation

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs- und Automatisierungstechnik, das mit den geltenden nationalen Normen vertraut ist.

Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.

Das Fachpersonal ist verpflichtet, stets die aktuell gültige Dokumentation zu verwenden.

Das Fachpersonal hat sicherzustellen, dass die Anwendung bzw. der Einsatz der beschriebenen Produkte alle Sicherheitsanforderungen, einschließlich sämtlicher anwendbaren Gesetze, Vorschriften, Bestimmungen und Normen erfüllt.

### Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte werden jedoch ständig weiterentwickelt.

Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.

Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

### Marken

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.

Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Bezeichnungen führen.

### Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.



EtherCAT® ist eine eingetragene Marke und patentierte Technologie lizenziert durch die Beckhoff Automation GmbH, Deutschland

### Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet.

Zu widerhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patent-, Gebrauchsmuster- oder Geschmacksmustereintragung vorbehalten.

## 1.2 Zu Ihrer Sicherheit

### Sicherheitsbestimmungen

Lesen Sie die folgenden Erklärungen zu Ihrer Sicherheit.  
Beachten und befolgen Sie stets produktspezifische Sicherheitshinweise, die Sie gegebenenfalls an den entsprechenden Stellen in diesem Dokument vorfinden.

### Haftungsausschluss

Die gesamten Komponenten werden je nach Anwendungsbestimmungen in bestimmten Hard- und Software-Konfigurationen ausgeliefert. Änderungen der Hard- oder Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig und bewirken den Haftungsausschluss der Beckhoff Automation GmbH & Co. KG.

### Qualifikation des Personals

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs-, Automatisierungs- und Antriebstechnik, das mit den geltenden Normen vertraut ist.

### Signalwörter

Im Folgenden werden die Signalwörter eingeordnet, die in der Dokumentation verwendet werden. Um Personen- und Sachschäden zu vermeiden, lesen und befolgen Sie die Sicherheits- und Warnhinweise.

### Warnungen vor Personenschäden

#### **GEFAHR**

Es besteht eine Gefährdung mit hohem Risikograd, die den Tod oder eine schwere Verletzung zur Folge hat.

#### **WARNUNG**

Es besteht eine Gefährdung mit mittlerem Risikograd, die den Tod oder eine schwere Verletzung zur Folge haben kann.

#### **VORSICHT**

Es besteht eine Gefährdung mit geringem Risikograd, die eine mittelschwere oder leichte Verletzung zur Folge haben kann.

### Warnung vor Umwelt- oder Sachschäden

#### **HINWEIS**

Es besteht eine mögliche Schädigung für Umwelt, Geräte oder Daten.

### Information zum Umgang mit dem Produkt



Diese Information beinhaltet z. B.:  
Handlungsempfehlungen, Hilfestellungen oder weiterführende Informationen zum Produkt.

## 1.3 Hinweise zur Informationssicherheit

Die Produkte der Beckhoff Automation GmbH & Co. KG (Beckhoff) sind, sofern sie online zu erreichen sind, mit Security-Funktionen ausgestattet, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen. Trotz der Security-Funktionen sind die Erstellung, Implementierung und ständige Aktualisierung eines ganzheitlichen Security-Konzepts für den Betrieb notwendig, um die jeweilige Anlage, das System, die Maschine und die Netzwerke gegen Cyber-Bedrohungen zu schützen. Die von Beckhoff verkauften Produkte bilden dabei nur einen Teil des gesamtheitlichen Security-Konzepts. Der Kunde ist dafür verantwortlich, dass unbefugte Zugriffe durch Dritte auf seine Anlagen, Systeme, Maschinen und Netzwerke verhindert werden. Letztere sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn entsprechende Schutzmaßnahmen eingerichtet wurden.

Zusätzlich sollten die Empfehlungen von Beckhoff zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Informationssicherheit und Industrial Security finden Sie in unserem <https://www.beckhoff.de/secguide>.

Die Produkte und Lösungen von Beckhoff werden ständig weiterentwickelt. Dies betrifft auch die Security-Funktionen. Aufgrund der stetigen Weiterentwicklung empfiehlt Beckhoff ausdrücklich, die Produkte ständig auf dem aktuellen Stand zu halten und nach Bereitstellung von Updates diese auf die Produkte aufzuspielen. Die Verwendung veralteter oder nicht mehr unterstützter Produktversionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Hinweise zur Informationssicherheit zu Produkten von Beckhoff informiert zu sein, abonnieren Sie den RSS Feed unter <https://www.beckhoff.de/secinfo>.

## 2 Übersicht

Die TwinCAT Analytics Runtime stellt sicher, dass eine kontinuierliche Datenanalyse parallel zu den eigentlichen Maschinenanwendungen läuft. Der SPS-Code, der in die Analytics Runtime heruntergeladen wird, kann mit Hilfe des Engineering-Produkts TE3500 Analytics Workbench automatisch generiert werden. Optional ist die Generierung eines vollständigen Analytics Dashboards auf Basis der HTML5-basierten TwinCAT HMI möglich.

Für den Fall, das Sie bereits eigene Visualisierungen haben, die Sie gerne per OPC UA oder ADS an die kontinuierliche Datenanalyse anbinden wollen, bieten wir Ihnen zwei unterschiedliche Runtime Produkte an. TF3550 beinhaltet den HMI Server und das HMI Client Pack 3, während TF3551 als Runtime Base die HMI nicht beinhaltet.

### Komponenten TF3550

- SPS-Runtime (TC1200)
- Analytics PLC Library (TF3510)
- IoT-Konnektivität mit MQTT und HTTPS/Rest (TF6701 und TF6760)
- Analytics Storage Provider Library
- HMI Server (TF2000)
- HMI Client Pack 3 (TF2020)
- HMI Scope (TF2300)
- Scope Server (TF3300)

### Komponenten TF3551

- SPS Runtime (TC1200)
- Analytics PLC Library (TF3510)
- IoT Konnektivität mit MQTT und HTTPS/Rest (TF6701 und TF6760)
- Scope Server (TF3300)
- Analytics Storage Provider Library



---

Seit TwinCAT Build 4024.57 ist die Anzahl der zu analysierenden Quellen (Analytics Logger, Analytics Data Exchange API, IoT Data Agent und EK9160) nicht mehr limitiert, es müssen keine zusätzlichen Controller Packs lizenziert werden.

---

## 3 Installation

### 3.1 Systemanforderungen

Das Analytics Runtime-Setup ist ein Rundum-Setup. Es enthält eine aktuelle Version von TwinCAT 3.1 XAR und des TwinCAT HMI Servers.

Technische Daten	TF3550 TwinCAT 3 Analytics Runtime
Zielsystem	Windows 7/8/10

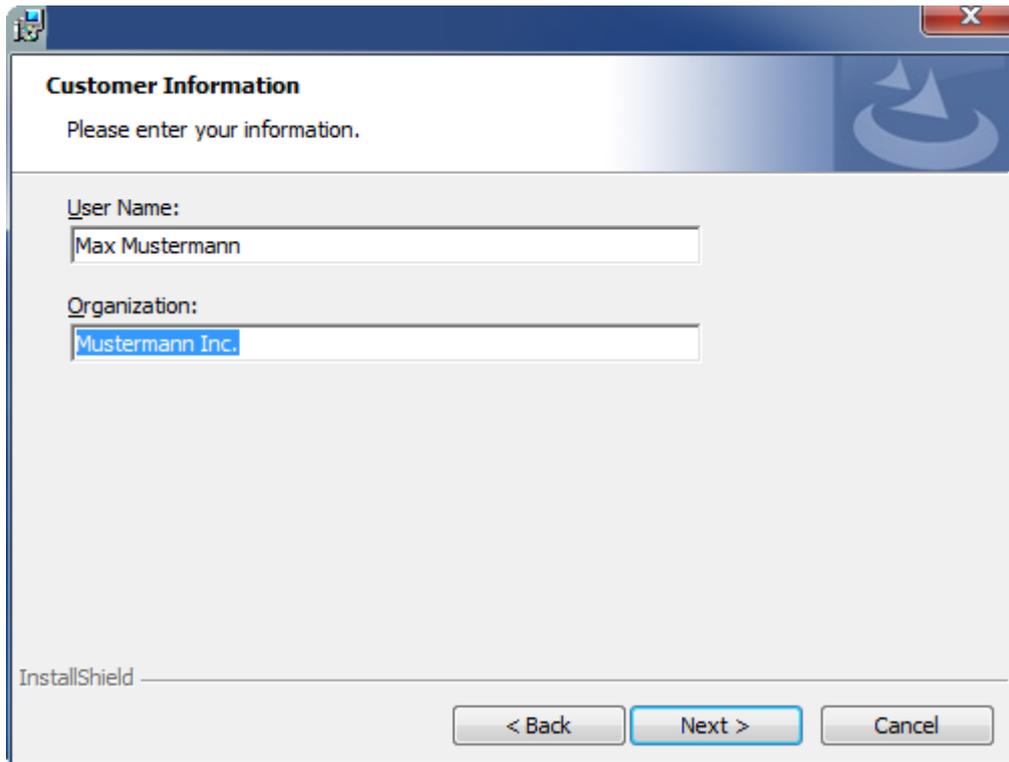
### 3.2 Installation

Nachfolgend wird beschrieben, wie die TwinCAT 3 Function für Windows-basierte Betriebssysteme installiert wird.

- ✓ Die Setup-Datei der TwinCAT 3 Function wurde von der Beckhoff-Homepage heruntergeladen.
- 1. Führen Sie die Setup-Datei als Administrator aus. Wählen Sie dazu im Kontextmenü der Datei den Befehl **Als Administrator ausführen**.
  - ⇒ Der Installationsdialog öffnet sich.
- 2. Akzeptieren Sie die Endbenutzerbedingungen und klicken Sie auf **Next**.



3. Geben Sie Ihre Benutzerdaten ein.



**Customer Information**  
Please enter your information.

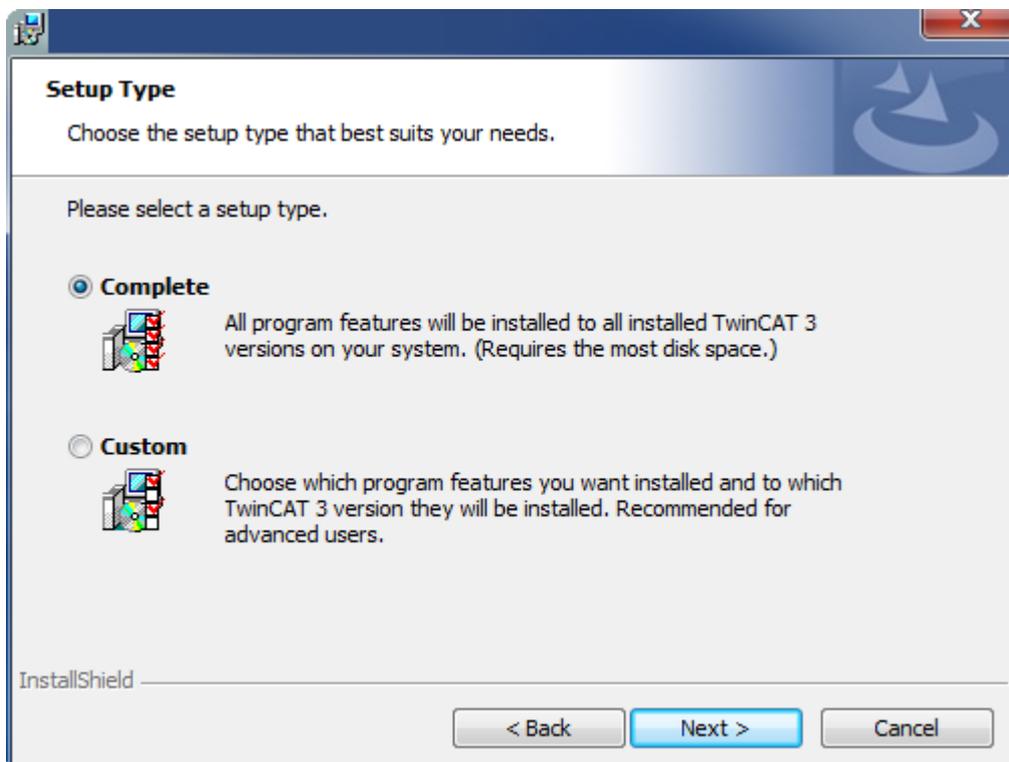
User Name:  
Max Mustermann

Organization:  
Mustermann Inc.

InstallShield

< Back   Next >   Cancel

4. Wenn Sie die TwinCAT 3 Function vollständig installieren möchten, wählen Sie **Complete** als Installationstyp. Wenn Sie die Komponenten der TwinCAT 3 Function separat installieren möchten, wählen Sie **Custom**.



**Setup Type**  
Choose the setup type that best suits your needs.

Please select a setup type.

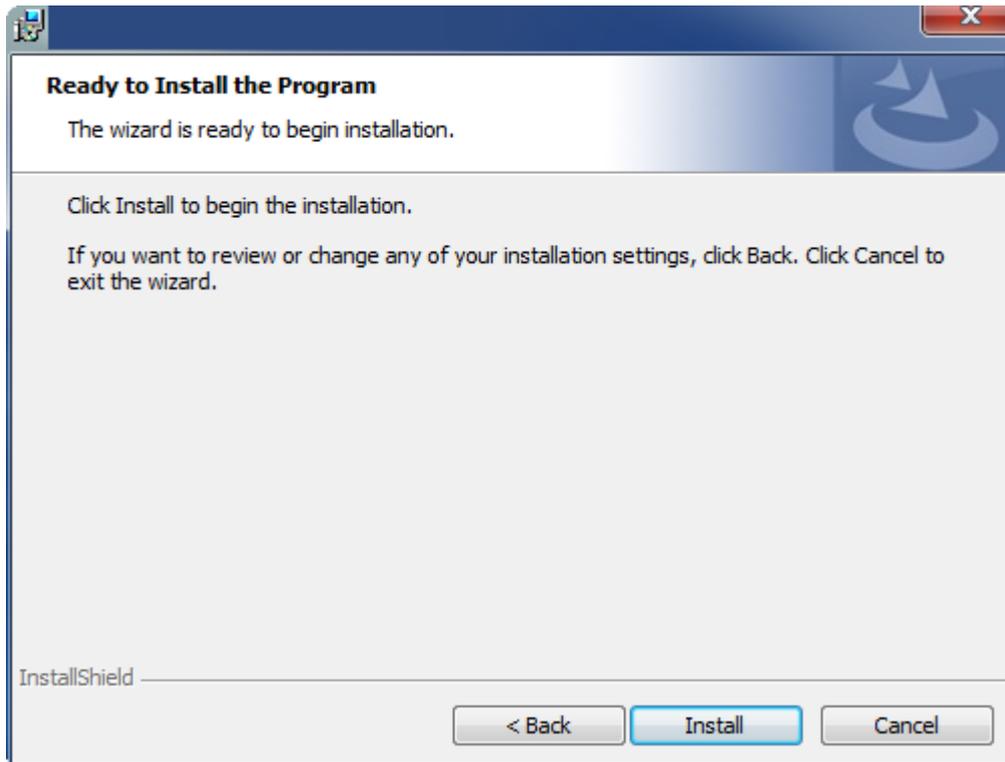
**Complete**  
All program features will be installed to all installed TwinCAT 3 versions on your system. (Requires the most disk space.)

**Custom**  
Choose which program features you want installed and to which TwinCAT 3 version they will be installed. Recommended for advanced users.

InstallShield

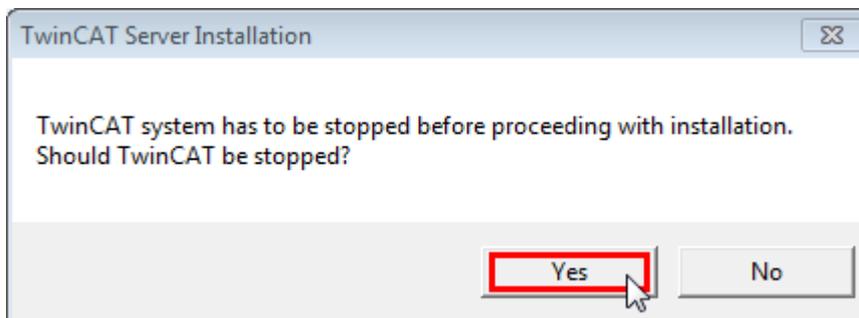
< Back   Next >   Cancel

5. Wählen Sie **Next** und anschließend **Install**, um die Installation zu beginnen.

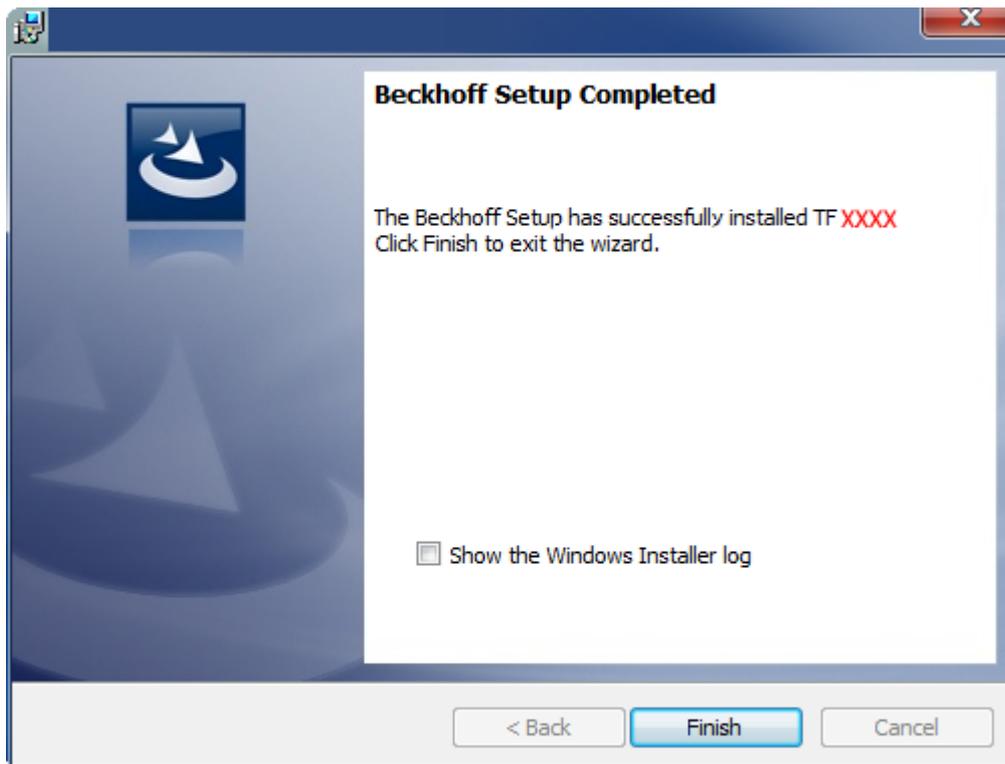


⇒ Ein Dialog weist Sie darauf hin, dass das TwinCAT-System für die weitere Installation gestoppt werden muss.

6. Bestätigen Sie den Dialog mit **Yes**.



7. Wählen Sie **Finish**, um das Setup zu beenden.



⇒ Die TwinCAT 3 Function wurde erfolgreich installiert.

### 3.3 Lizenzierung

Die TwinCAT 3 Function ist als Vollversion oder als 7-Tage-Testversion freischaltbar. Beide Lizenztypen sind über die TwinCAT-3-Entwicklungsumgebung (XAE) aktivierbar.

#### Lizenzierung der Vollversion einer TwinCAT 3 Function

Die Beschreibung der Lizenzierung einer Vollversion finden Sie im Beckhoff Information System in der Dokumentation „[TwinCAT-3-Lizenzierung](#)“.

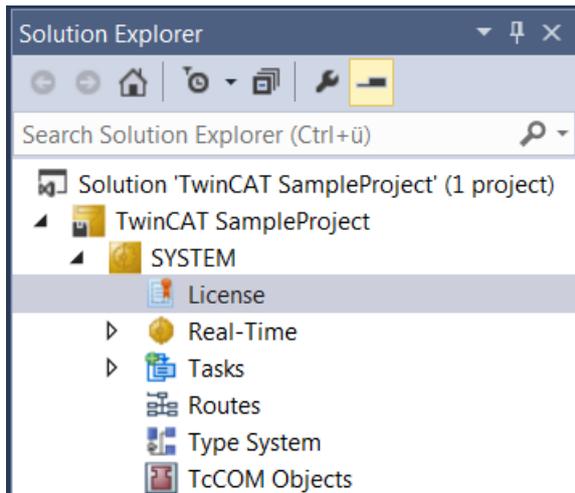
#### Lizenzierung der 7-Tage-Testversion einer TwinCAT 3 Function



Eine 7-Tage-Testversion kann nicht für einen [TwinCAT-3-Lizenz-Dongle](#) freigeschaltet werden.

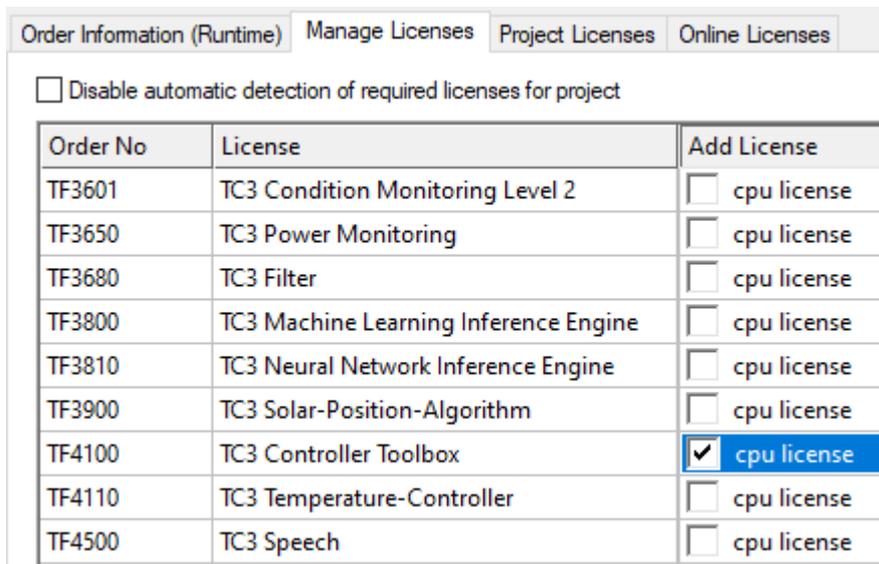
1. Starten Sie die TwinCAT-3-Entwicklungsumgebung (XAE).
2. Öffnen Sie ein bestehendes TwinCAT-3-Projekt oder legen Sie ein neues Projekt an.
3. Wenn Sie die Lizenz für ein Remote-Gerät aktivieren wollen, stellen Sie das gewünschte Zielsystem ein. Wählen Sie dazu in der Symbolleiste in der Drop-down-Liste **Choose Target System** das Zielsystem aus.
  - ⇒ Die Lizenzierungseinstellungen beziehen sich immer auf das eingestellte Zielsystem. Mit der Aktivierung des Projekts auf dem Zielsystem werden automatisch auch die zugehörigen TwinCAT-3-Lizenzen auf dieses System kopiert.

4. Klicken Sie im **Solution Explorer** im Teilbaum **SYSTEM** doppelt auf **License**.



⇒ Der TwinCAT-3-Lizenzmanager öffnet sich.

5. Öffnen Sie die Registerkarte **Manage Licenses**. Aktivieren Sie in der Spalte **Add License** das Auswahlkästchen für die Lizenz, die Sie Ihrem Projekt hinzufügen möchten (z. B. „TF4100 TC3 Controller Toolbox“).



6. Öffnen Sie die Registerkarte **Order Information (Runtime)**.

⇒ In der tabellarischen Übersicht der Lizenzen wird die zuvor ausgewählte Lizenz mit dem Status „missing“ angezeigt.

7. Klicken Sie auf **7 Days Trial License...**, um die 7-Tage-Testlizenz zu aktivieren.

The screenshot shows the 'License Management' window with the following sections:

- Order Information (Runtime)**: Includes tabs for 'Manage Licenses', 'Project Licenses', and 'Online Licenses'. Below are fields for 'License Device' (set to 'Target (Hardware Id)'), 'System Id' (2DB25408-B4CD-81DF-5488-6A3D9B49EF19), and 'Platform' (other (91)).
- License Request**: Includes a 'Provider' dropdown set to 'Beckhoff Automation', a 'Generate File...' button, and input fields for 'License Id', 'Customer Id', and 'Comment'.
- License Activation**: This section is highlighted with a red box and contains two buttons: '7 Days Trial License...' and 'License Response File...'.

⇒ Es öffnet sich ein Dialog, der Sie auffordert, den im Dialog angezeigten Sicherheitscode einzugeben.

The 'Enter Security Code' dialog box contains the following elements:

- Title: 'Enter Security Code' with a close button (X).
- Text: 'Please type the following 5 characters:'
- Code display: A box showing the code 'Kg8T4'.
- Input field: A two-character input field with a red border, currently empty.
- Buttons: 'OK' (highlighted with a red box) and 'Cancel'.

8. Geben Sie den Code genauso ein, wie er angezeigt wird, und bestätigen Sie ihn.

9. Bestätigen Sie den nachfolgenden Dialog, der Sie auf die erfolgreiche Aktivierung hinweist.

⇒ In der tabellarischen Übersicht der Lizenzen gibt der Lizenzstatus nun das Ablaufdatum der Lizenz an.

10. Starten Sie das TwinCAT-System neu.

⇒ Die 7-Tage-Testversion ist freigeschaltet.

## 4 Analytics Workflow – Erste Schritte

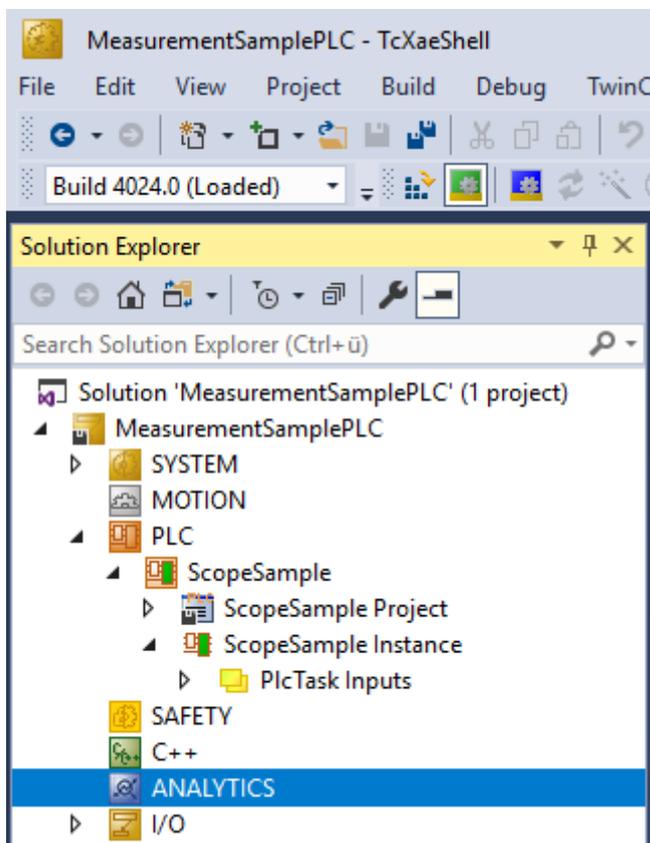
Diese Dokumentation präsentiert Schritt für Schritt den vollständigen TwinCAT Analytics Workflow. Von der Datenerfassung über die Kommunikation und Historisierung bis hin zur Auswertung und Analyse der Daten sowie zur Präsentation der Daten im webbasierten Dashboard.

### 4.1 Aufzeichnung von Maschinendaten

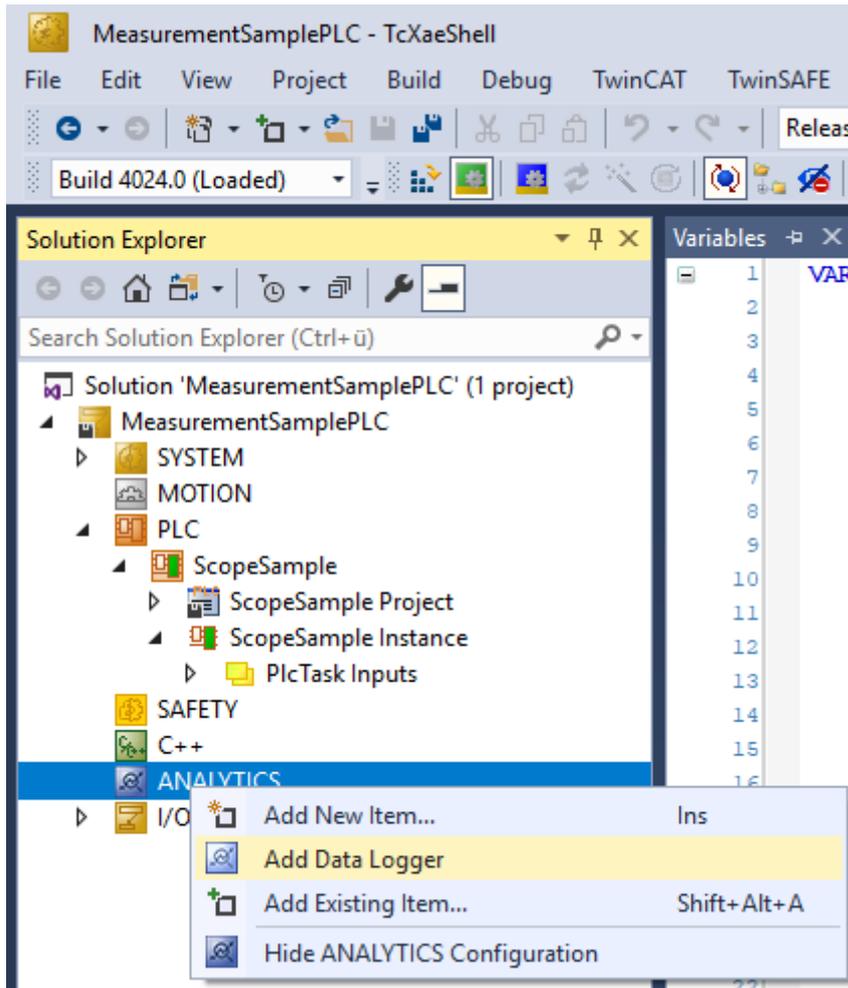
Maschinenseitig ist der Analytics Logger die Aufzeichnungseinrichtung von Prozessdaten des Maschinenabbaus, SPS, NC usw. Der Logger arbeitet im Echtzeitkontext von TwinCAT.

Der TwinCAT Analytics Logger wird mit TwinCAT XAE und XAR installiert. Der Logger kann als MQTT-Client fungieren, um die aufgezeichneten Daten an einen nativen MQTT-Message-Broker zu übermitteln, oder die Daten im selben Datenformat in einer lokalen Binärdatei zu speichern. Bei Verwendung als MQTT-Client kann der Logger kurze Unterbrechungen der Verbindung zum Message-Broker mit Hilfe einer Ringpuffer-Funktionalität überbrücken. Auch für den lokalen Binärdateispeicher kann ein Ringpuffer konfiguriert werden.

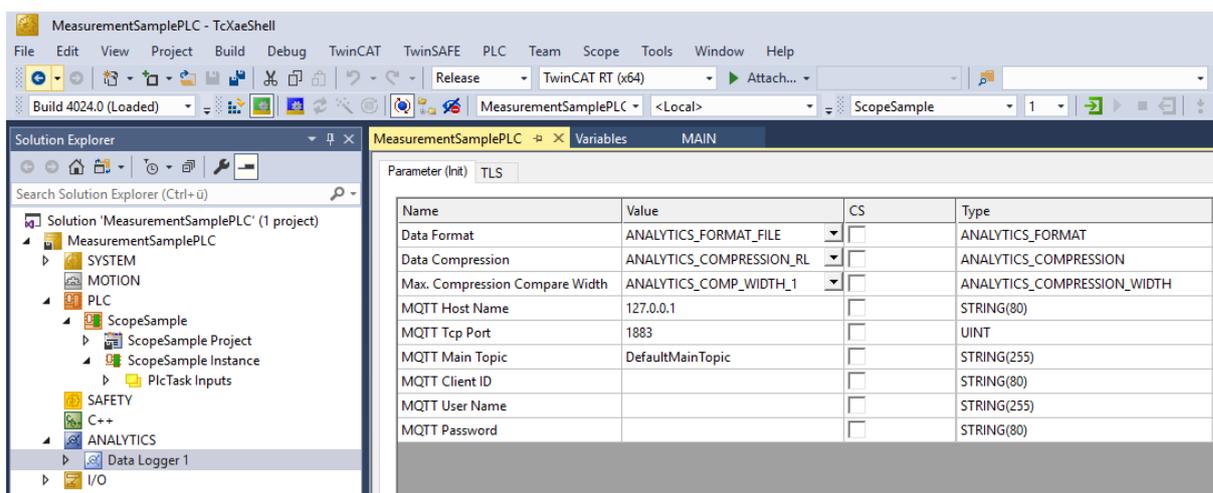
- Um den Analytics Logger zu konfigurieren, müssen Sie in Ihrem vorhandenen TwinCAT-Projekt zum Analytics-Baumknoten navigieren.



- Klicken Sie mit der rechten Maustaste auf diesen Knoten und klicken Sie auf **Add Data Logger**, um Ihrer Konfiguration eine neue Instanz hinzuzufügen.



- Für die Konfiguration der Grundeinstellungen doppelklicken Sie auf das neue Tree Item.



Sie können Ihre spezifischen Analytics Logger-Einstellungen vornehmen.

-Datenformat: Binärdatei oder MQTT-Stream.

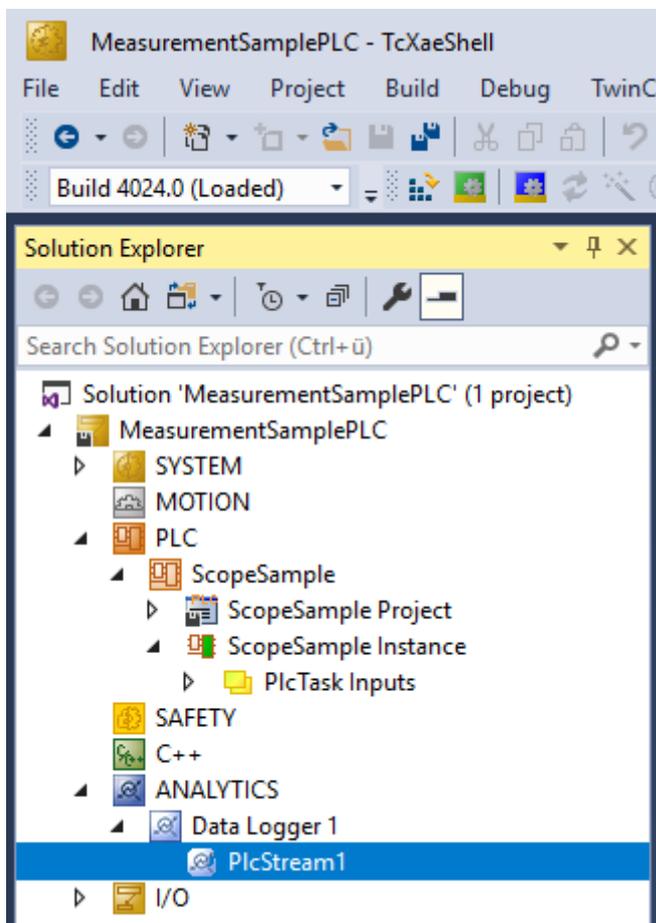
-FILE-Format: Der Analytics Logger speichert die Daten in lokalen Binärdateien und alle anderen Einstellungen sind nicht mehr notwendig. Die Dateien werden in `C:\TwinCAT\3.1\Boot\Analytics` gespeichert.

-BINARY-Format: Die Daten werden an den konfigurierten MQTT-Message-Broker gesendet. Sie können mehrere Logger in einem TwinCAT-Projekt vorsehen, um Daten an verschiedene MQTT-Message-Broker zu übermitteln.

- Datenkompression: ein (Standard) oder aus.
- Max. Kompression: Modus der Kompression.
- MQTT-Hostname
- MQTT-Tcp-Port
- MQTT-Maintopic für eigene Hierarchieebenen, damit die Identifikation einfach bleibt.
- MQTT-Client-ID, sollte im Netzwerk eindeutig sein.
- MQTT-Benutzername
- MQTT-Passwort für die Authentifizierung beim Message-Broker.

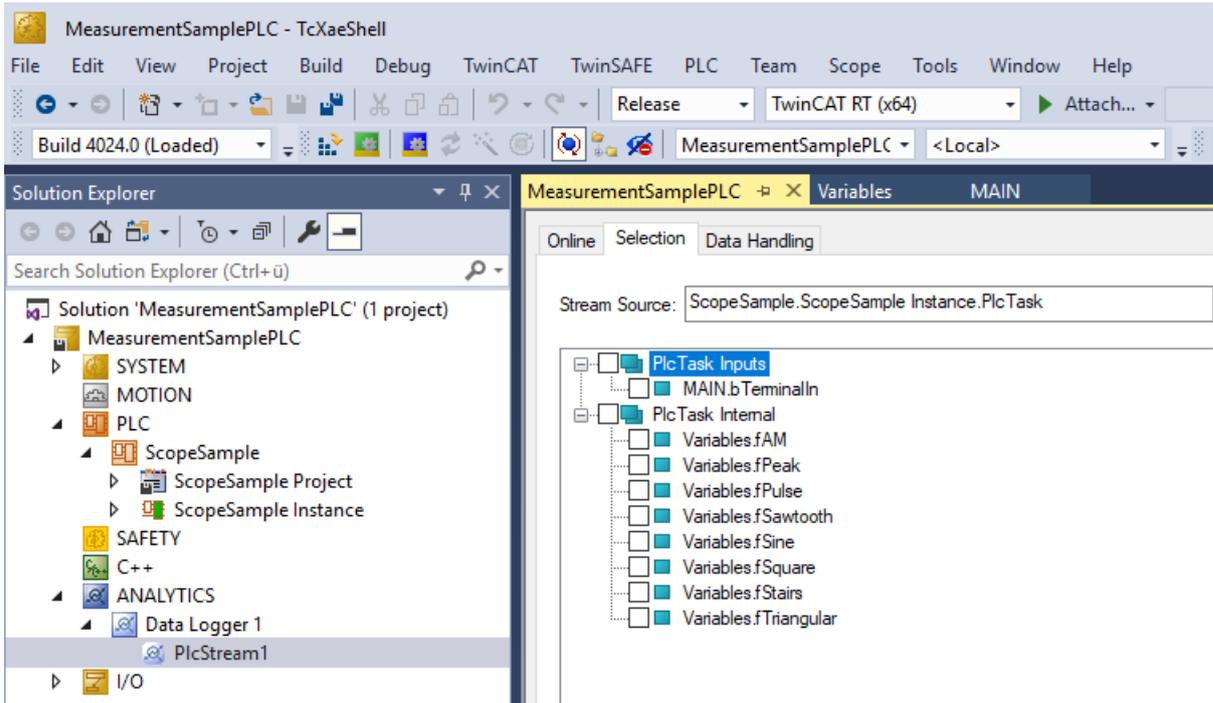
-Auf der Registerkarte **TLS** (Transport Layer Security) können die Sicherheitseinstellungen konfiguriert werden. TLS ist ein sicherer Kommunikationskanal zwischen Client und Server. Bei Verwendung von Zertifikaten ist der TCP-Port 8883 ausschließlich für MQTT über TLS reserviert! Der Analytics Logger unterstützt die Modi CA Certificates, CA Certificates & Client Certificate und Preshared Key (PSK).

- Wenn Variablen in Ihrer SPS-Anwendung in der Deklaration mit dem Attribut {attribute 'TcAnalytics'} gekennzeichnet sind, werden sie automatisch als Stream unter dem Daten-Logger-Baumknoten angezeigt.

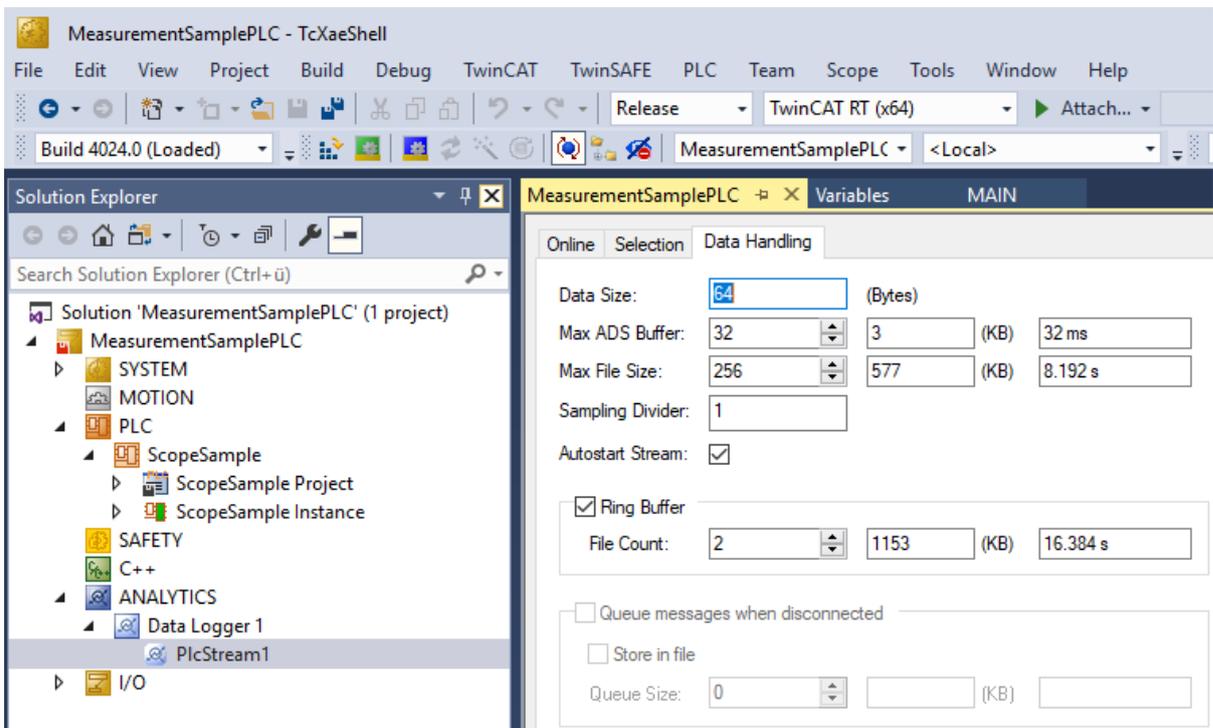


Ein zusätzlicher Geräte-Stream wird angezeigt, wenn Ihre Konfiguration ein EtherCAT-Prozessabbild liefert.

- In dem Stream gibt es einen Karteireiter **Selection**, wo die aufzuzeichnenden Variablen ausgewählt werden können.

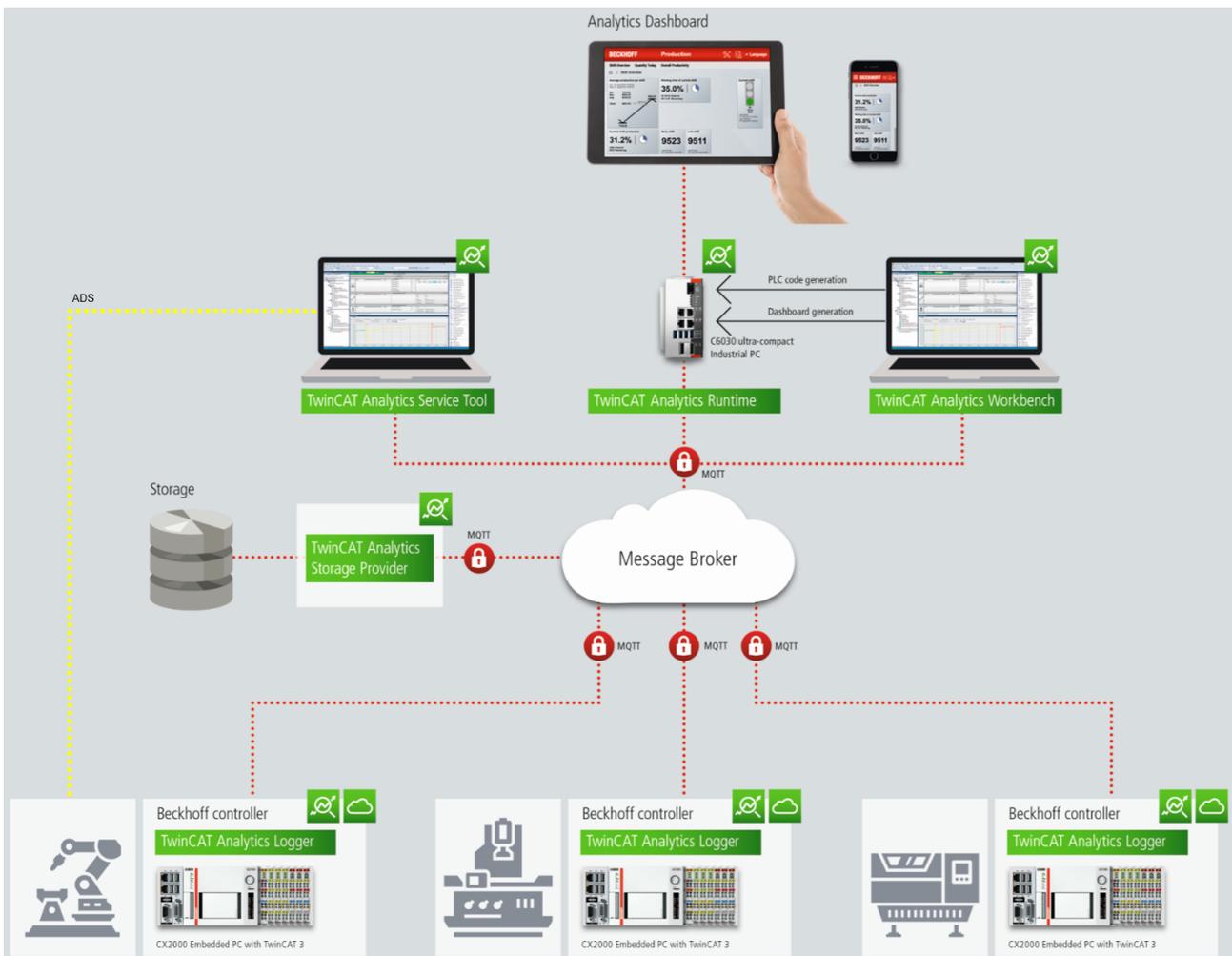


- Schließlich ist es im Karteireiter **Data Handling** möglich, die Paketgröße für die Frames zu ändern oder den Ringpuffer für Verbindungsunterbrechungen und Datei zu konfigurieren.



## 4.2 Kommunikation

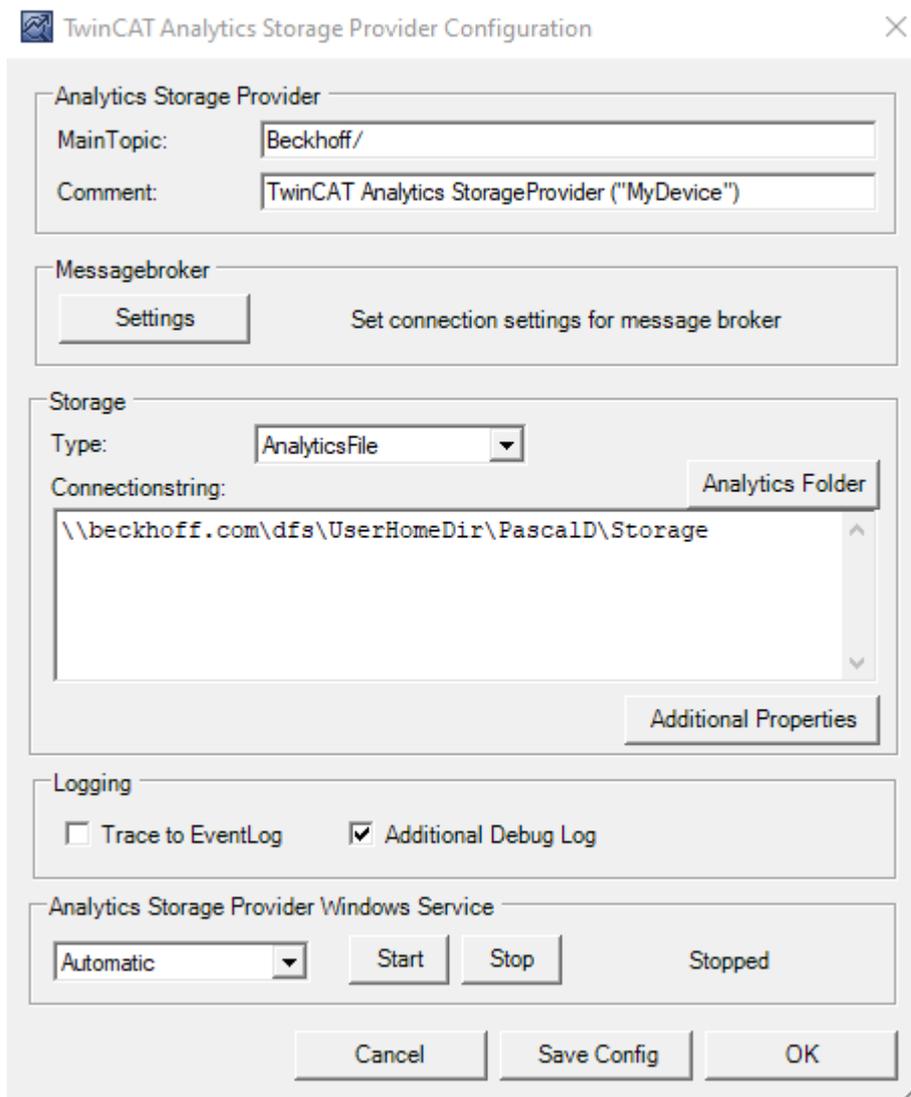
Derzeit kann der Analytics-Workflow vollständig über MQTT abgebildet werden. Die Engineering-Tools können auch über ADS auf die Daten der Maschinen zugreifen und Analysen durchführen.



Wenn man für das IoT-Kommunikationsprotokoll MQTT wählt, muss irgendwo im Netzwerk ein nativer MQTT-Message-Broker eingerichtet werden (VM in einem Cloud-System ist ebenfalls möglich). Dieser Message-Broker sorgt für eine Entkopplung der verschiedenen Anwendungen im Analytics-Workflow.

### 4.3 Historisierung von Daten

Nach der Installation des TwinCAT Analytics Storage Providers kann der im Hintergrund laufende Dienst konfiguriert werden. Hierzu finden Sie die Anwendung `TwinCAT.Analytics.StorageProvider.Configurator` im Ordner `C:\TwinCAT\Functions\TF3520-Analytics-StorageProvider\Tools`.

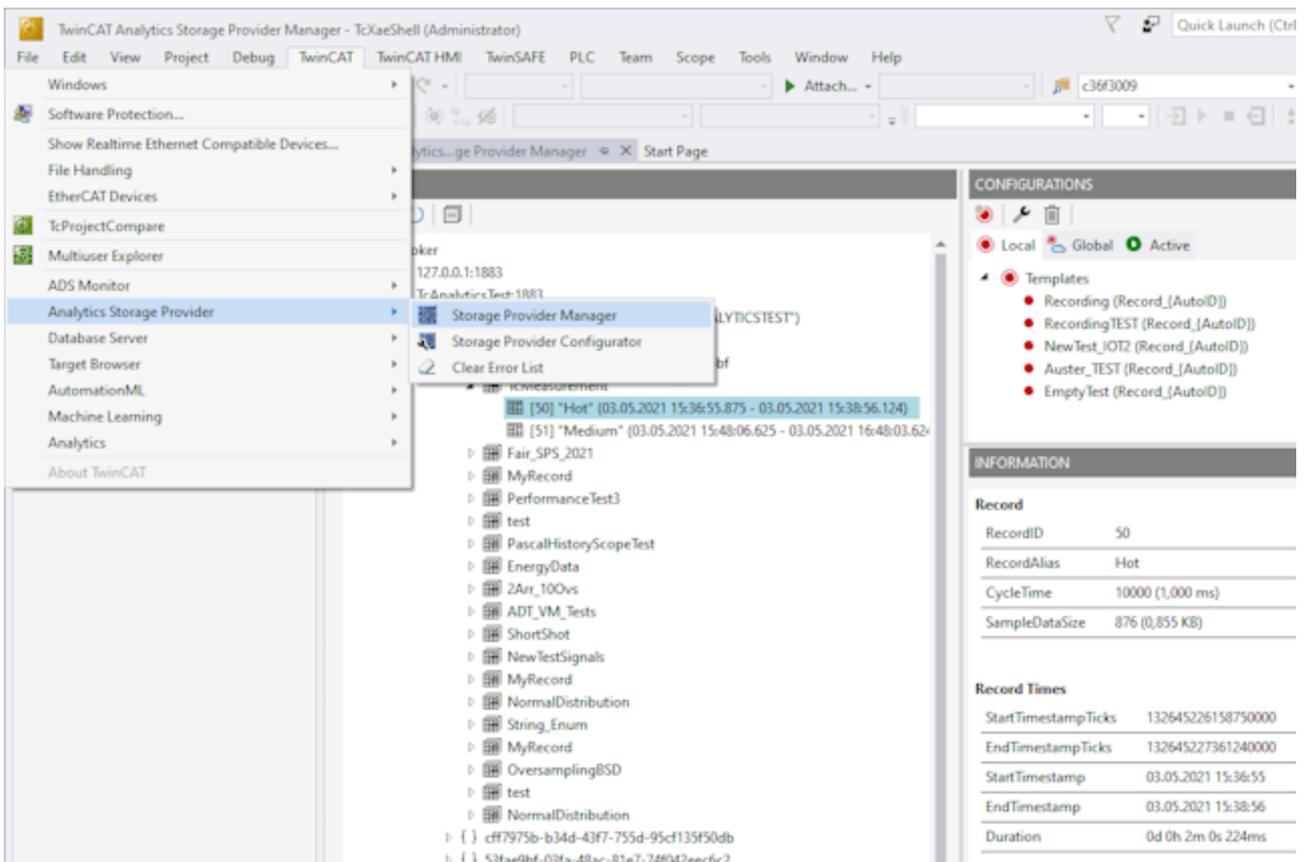


Der Hauptteil des Topics kann in der Konfiguration ebenso festgelegt werden wie der Kommentar, der zur Identifizierung dient, wenn mehr als ein Storage Provider beim Message-Broker registriert ist.

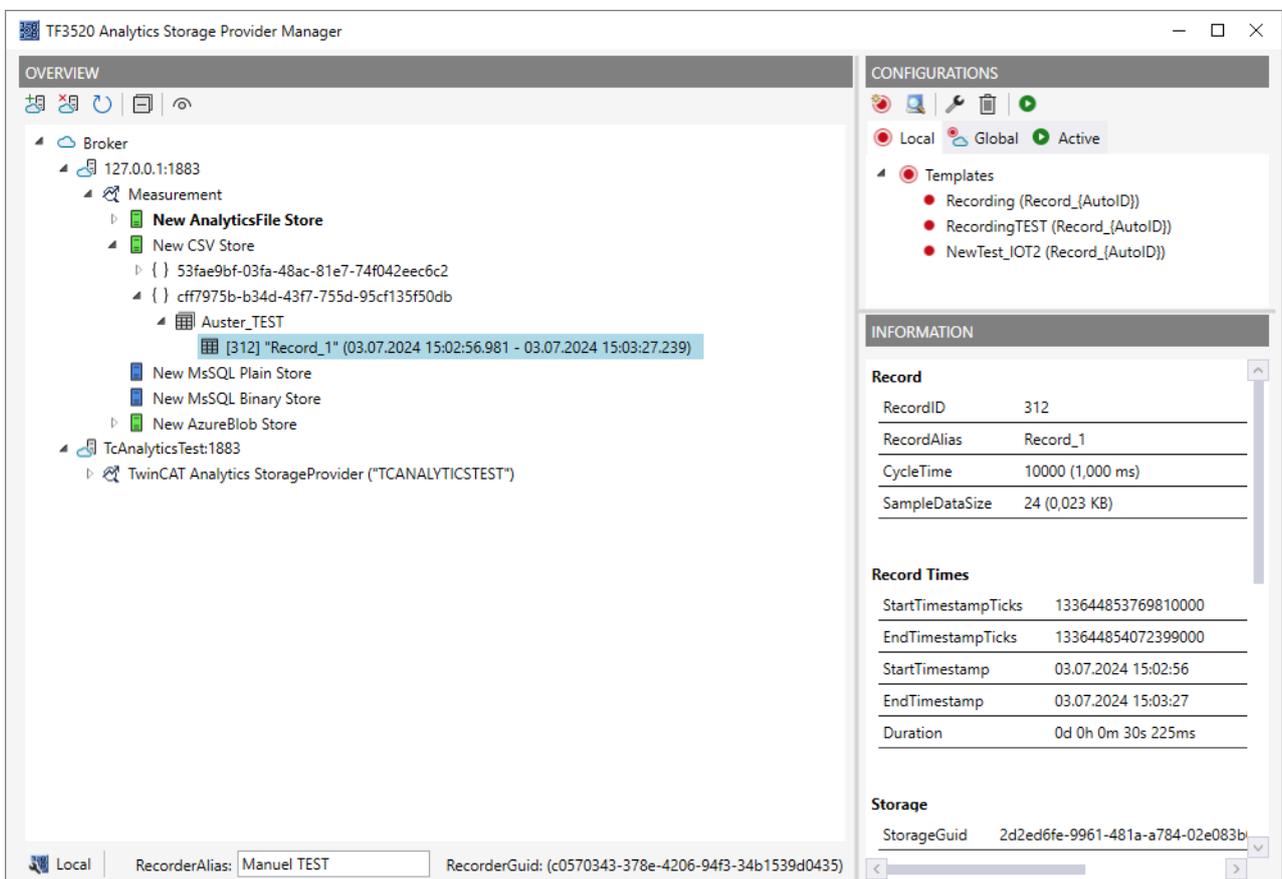
Sie können die Message-Broker-Einstellungen vornehmen und sich für einen Speichertyp entscheiden:

- Analytics File (Binärdatei)
- CSV Datei
- Microsoft SQL (Binär / Klartext)
- InfluxDB (Klartext)
- Microsoft Azure Blob (Azure-Cloud erforderlich)

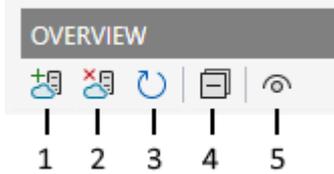
Zuletzt können Sie die Konfiguration speichern und den Dienst starten. Der nächste Schritt besteht in der Konfiguration der spezifischen Aufzeichnung. Hierzu sollten Sie in Ihrer Entwicklungsumgebung den **Storage Provider Manager** auswählen.



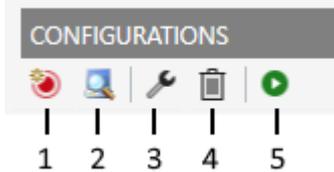
Mit dem Storage Provider Recorder können Aufnahme Definitionen erstellt, gestartet und verwaltet werden. Zusätzlich besteht die Möglichkeit, die Datenspeicher einzelner Analytics Storage Provider zu verwalten. Alle wichtigen Eigenschaften der gefundenen Analytics Storage Providern und historisierten Daten werden übersichtlich dargestellt.



**Symbolleisten**



1	Neuen Broker hinzufügen
2	Ausgewählten Broker entfernen
3	Anzeige aktualisieren
4	Alle Knoten einklappen
5	Ansicht Wechsel zwischen Dark/Light Modus

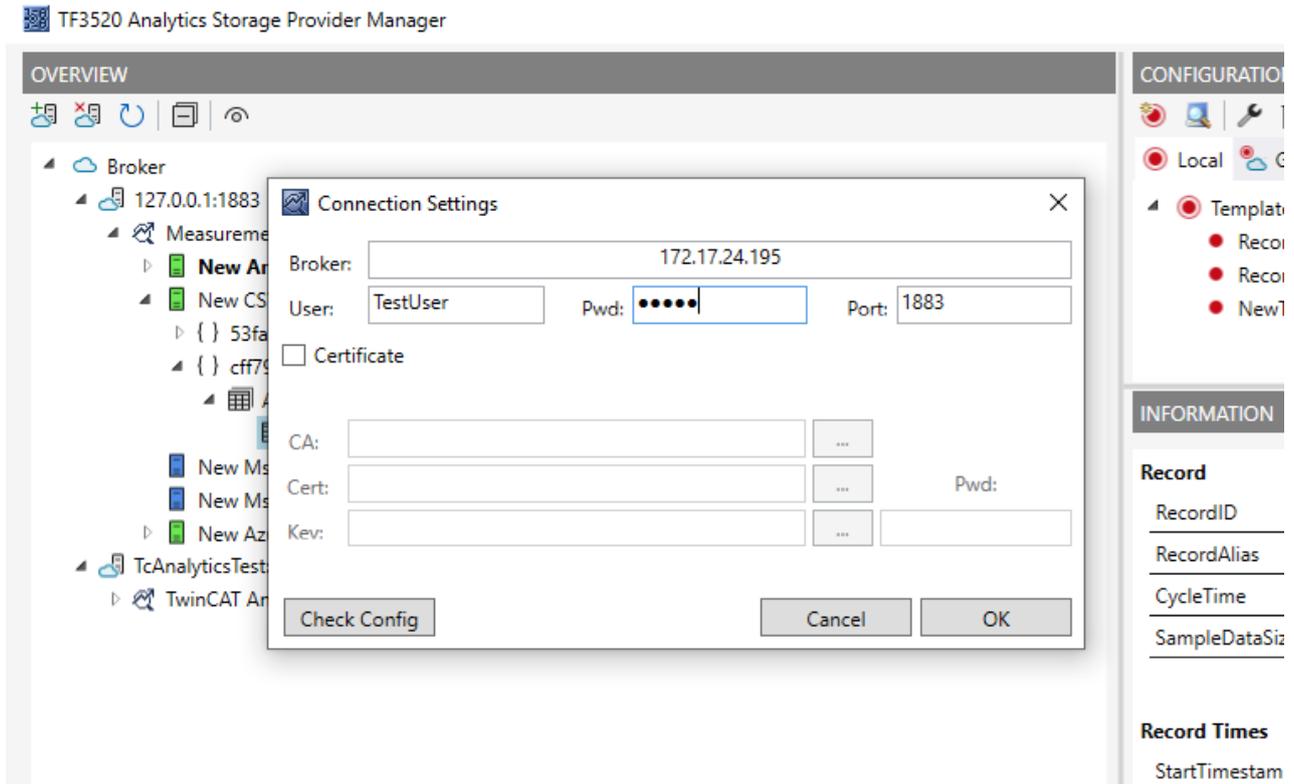


1	Neue Aufnahme Definition erzeugen
2	Target Browser öffnen
3	Ausgewählte Definition bearbeiten
4	Ausgewählte Definition löschen
5	Ausgewählte Definition starten

**Manager Fenster („OVERVIEW“) einrichten**

Vergeben Sie zunächst einen „RecorderAlias“. Dieser hilft die gestarteten Aufnahmen zu gruppieren und seine selbst gestarteten wieder zu finden.

Danach können ein oder mehrere Broker eingerichtet werden. Dies geschieht über die schon bekannte Eingabemaske für MQTT-Verbindungseigenschaften.



Sobald eine Verbindung zum Broker hergestellt werden konnte, werden alle darauf verbundenen Analytics Storage Provider aufgelistet.

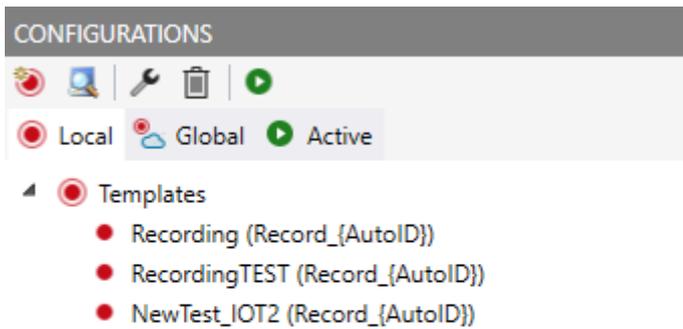
**„Storage“ Status**

- Broker
  - 127.0.0.1:1883
    - Measurement
      - 1 - New AnalyticsFile Store
      - 2 - New CSV Store
      - 3 - New InfluxDB Store
      - 4 - New MySQL Plain Store
      - 5 - New MySQL Binary Store
      - 6 - New AzureBlob Store

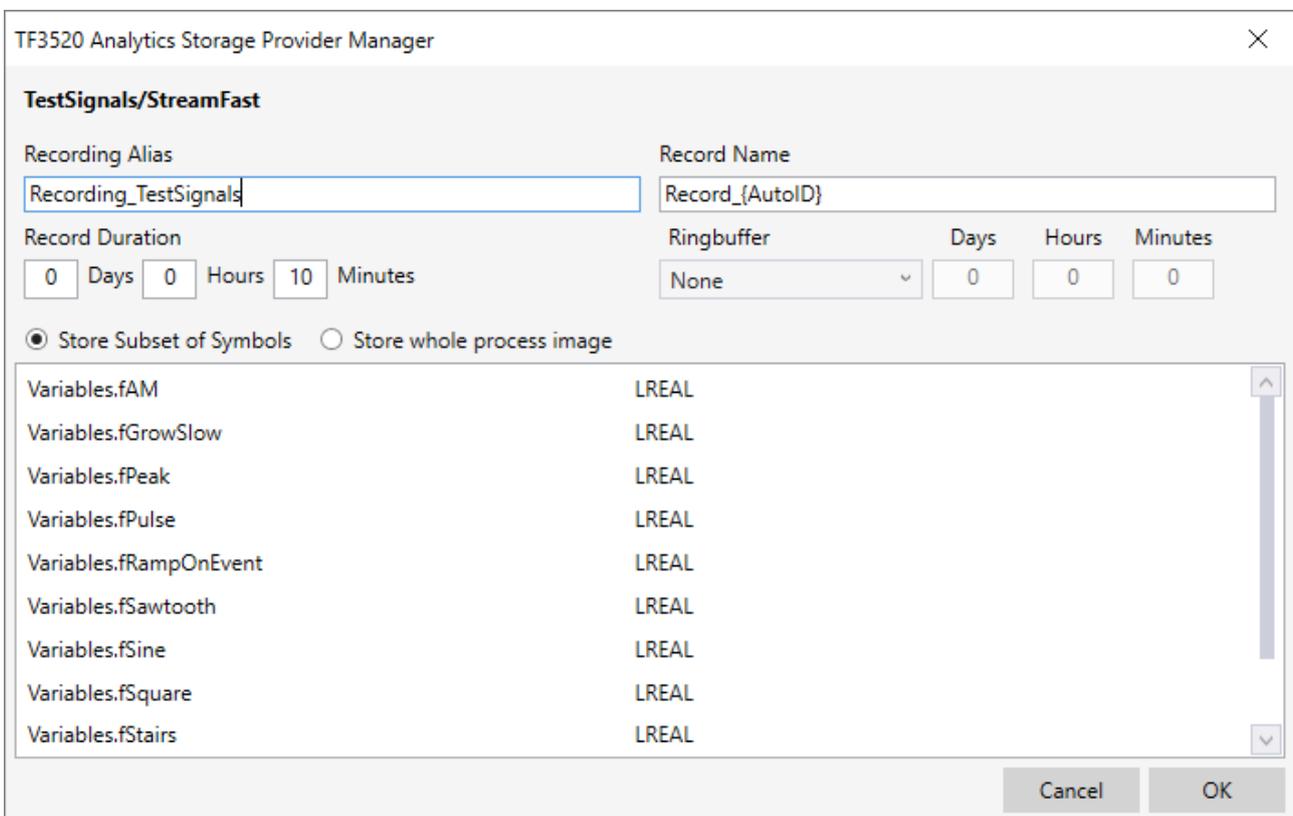
1	Storage Online
2	Storage Offline
3	Storage startet auf
4	Storage startet mit Fehler. Es wird weiter versucht ihn zu starten
5	Storage wird heruntergefahren
6	Storage befindet sich im Fehlerfall

**Aufnahmedefinitionen („CONFIGURATIONS“)**

Lokal:



Aufnahmedefinitionen die lokal gespeichert sind, werden im Tab „Local“ aufgelistet. Hier können auch neue Definitionen erzeugt werden. Hierzu drücken Sie den Button „Neue Definition erzeugen“. Folgender Dialog öffnet sich.



Sie können nun die Symbole, die aufgezeichnet werden sollen, vom Target Browser per Drag Drop in den Dialog ziehen. Zusätzlich vergeben sie ein Recording Alias und einen Record Name.

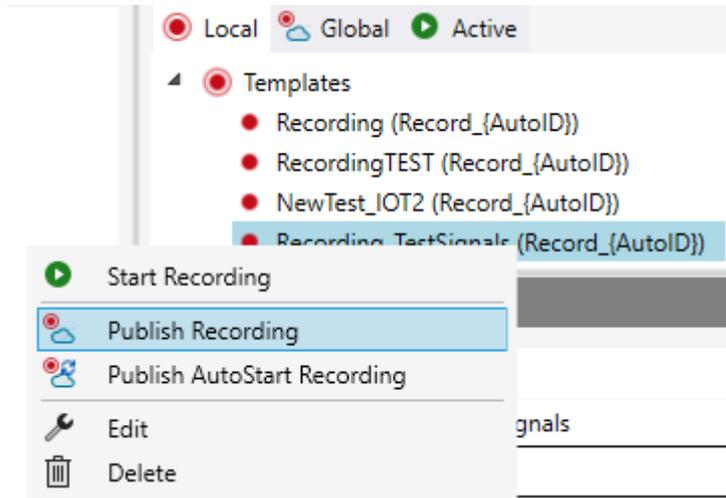
Für den Record Name stehen verschiedene Platzhalter zur Verfügung:

"{AutoID}"	
"{Topic}"	
"{SystemID}"	
"{Layout}"	
"{CycleTime}"	
"{SampleSize}"	
"{RecordStart}"	

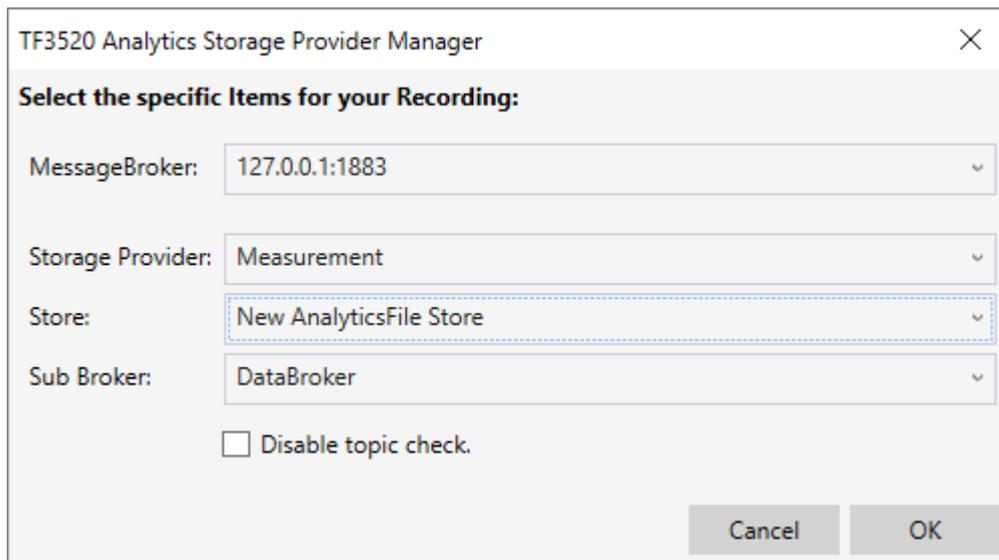
Sie können auch Aufzeichnungsnamen und eine Dauer konfigurieren (anderenfalls läuft die Aufzeichnung endlos, bis sie manuell gestoppt wird). Ein Ringpuffer kann nach Storage oder Zeit eingestellt werden.

Mit OK werden die Eingaben bestätigt und eine neue lokale Aufnahmedefinition wird erzeugt.

Es ist nun möglich diese Definition direkt über die Toolbar oder das Kontextmenü zu starten. Es besteht aber auch die Möglichkeit die Definition global zugänglich zu machen. Dies kann über das Kontextmenü mit dem Eintrag „Publish Recording“ gemacht werden.

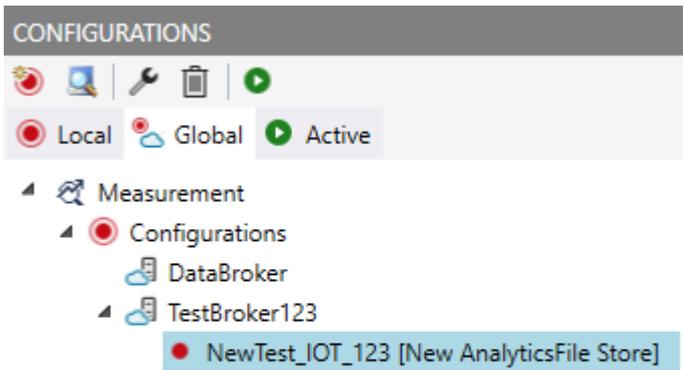


Danach öffnet sich folgender Dialog:



Hier kann nun der gewünschte Analytics Storage Provider ausgewählt werden, über den die Definition veröffentlicht werden soll. Zusätzlich wird der Definition ein Storage und ein Sub Broker des ausgewählten Analytics Storage Providers zugewiesen. Nach der Auswahl wird die Aufnahmedefinition mit OK bestätigt und an dem ausgewählten Analytics Storage Provider veröffentlicht. Dadurch kann sie von jedem Storage Provider Manager, der mit dem MQTT-Broker verbunden ist, unter dem „Global“ Tab gefunden werden.

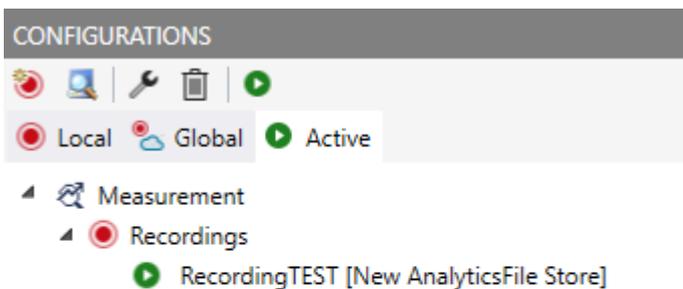
Global:



Hier werden alle globalen Aufnahmedefinitionen des ausgewählten Analytics Storage Providers aufgelistet. Die Definitionen werden den Sub Brokern zugeordnet. Diesen globalen Definitionen können verändert, heruntergeladen und gestartet werden.

Gestartete Aufnahmedefinitionen werden im Tab „Active“ aufgelistet.

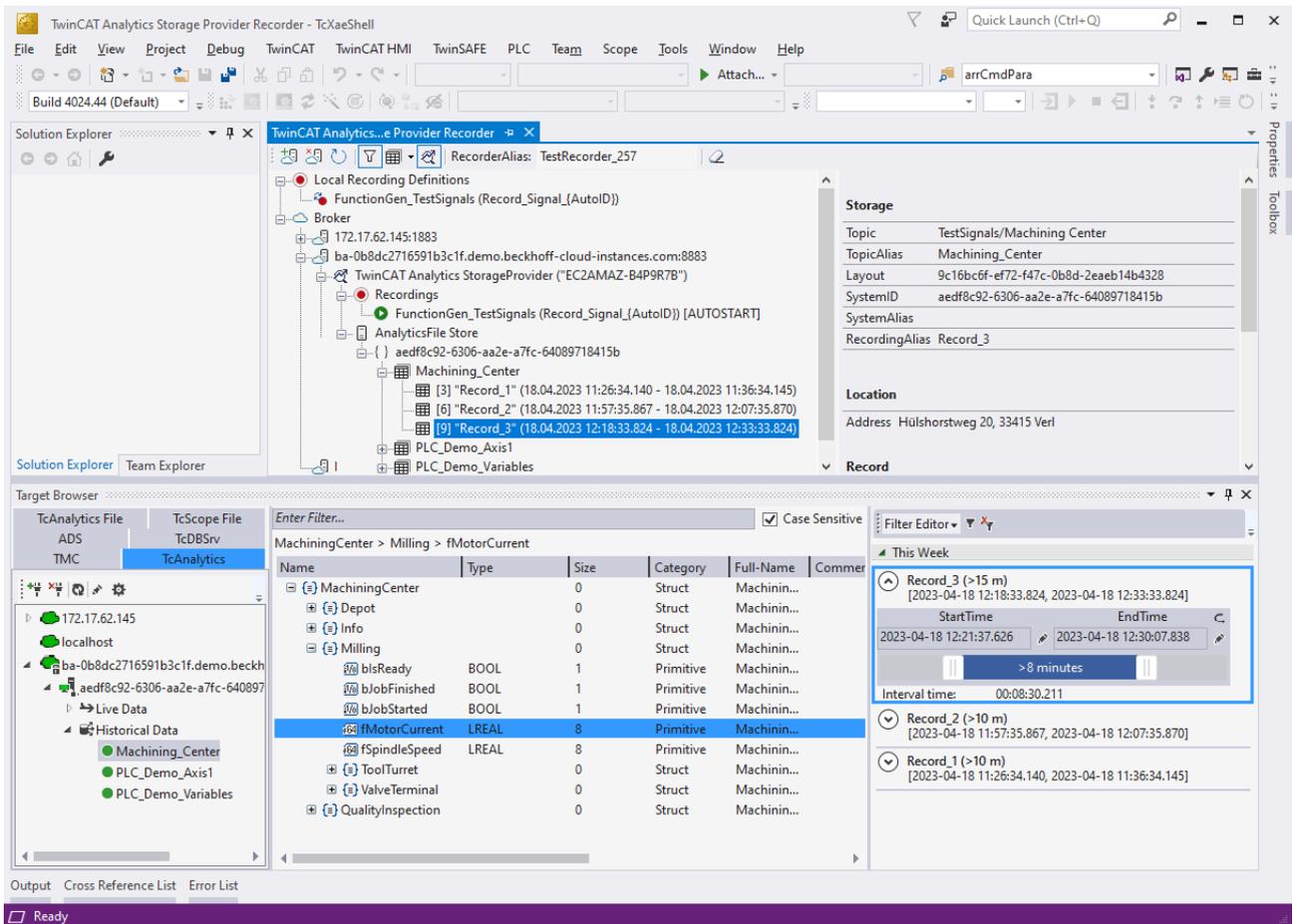
Aktiv:



Hier werden alle aktiven Recordings von allen Usern aufgelistet. Die Recordings können in diesem Tab beendet werden, es ist auch möglich zu dem resultierenden Record zu springen.

**Historisierte Daten verwenden**

Nach und auch während der Aufzeichnung können Sie die historischen Daten als Eingang für Ihre Analyse im Target Browser auswählen. Im Target Browser finden Sie für die historischen Daten eine neue Steuerung auf der rechten Seite. Dort können Sie die Zeitspanne für Ihre Daten auswählen.

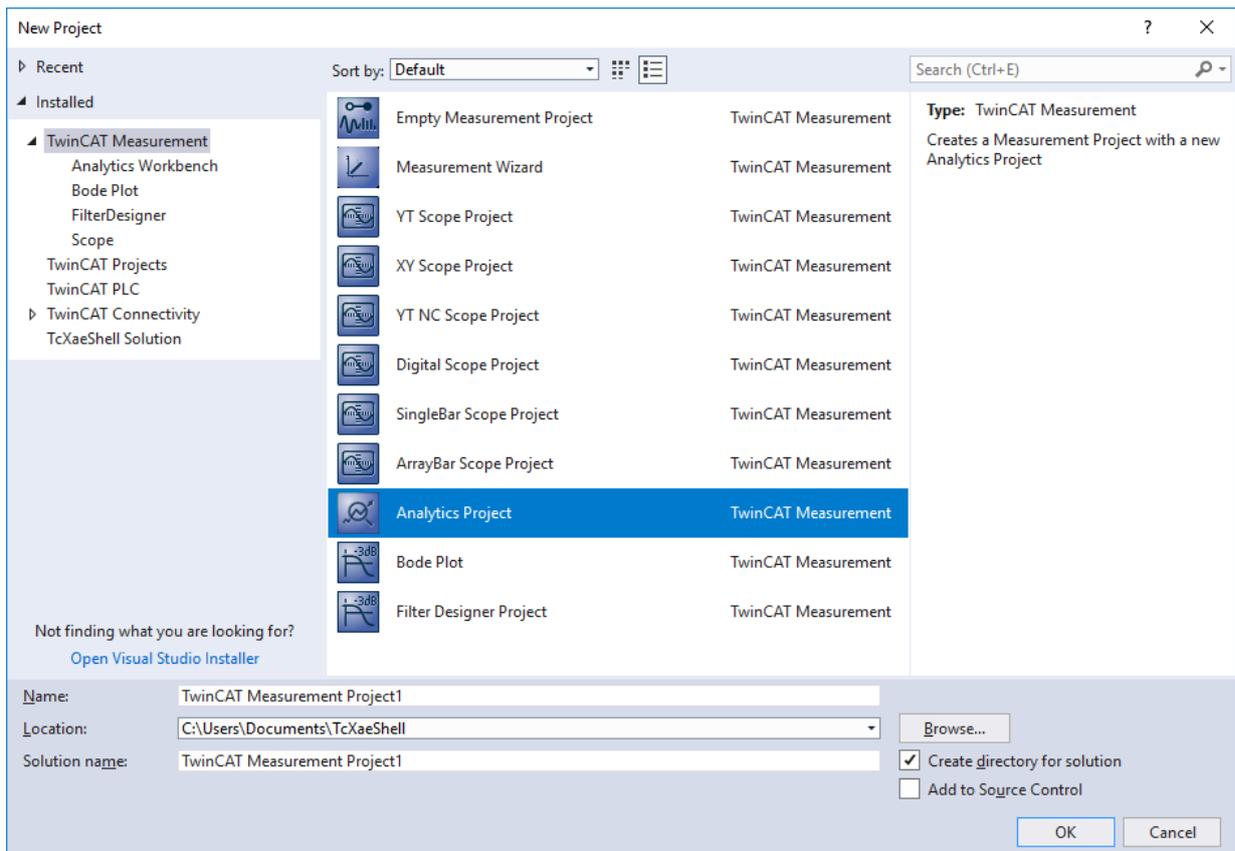


## 4.4 Analyse der Daten

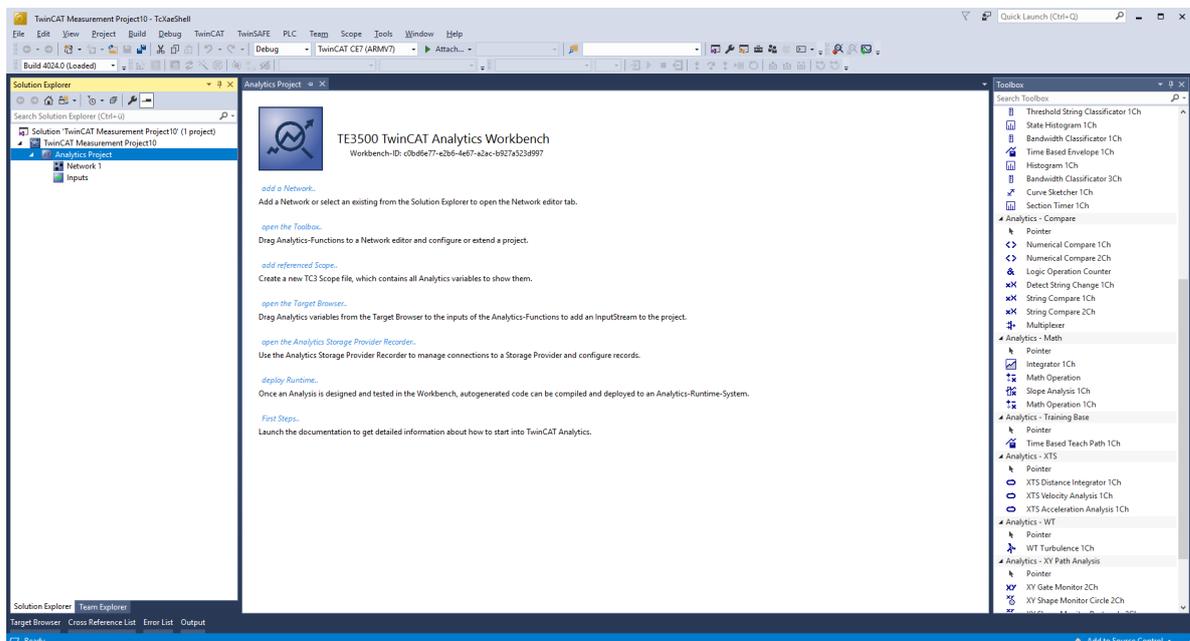
✓ Öffnen Sie Ihre TwinCAT Engineering-Umgebung, um die Analyse der Daten zu starten.

1. Öffnen Sie **Visual Studio® > File > New > Project...**

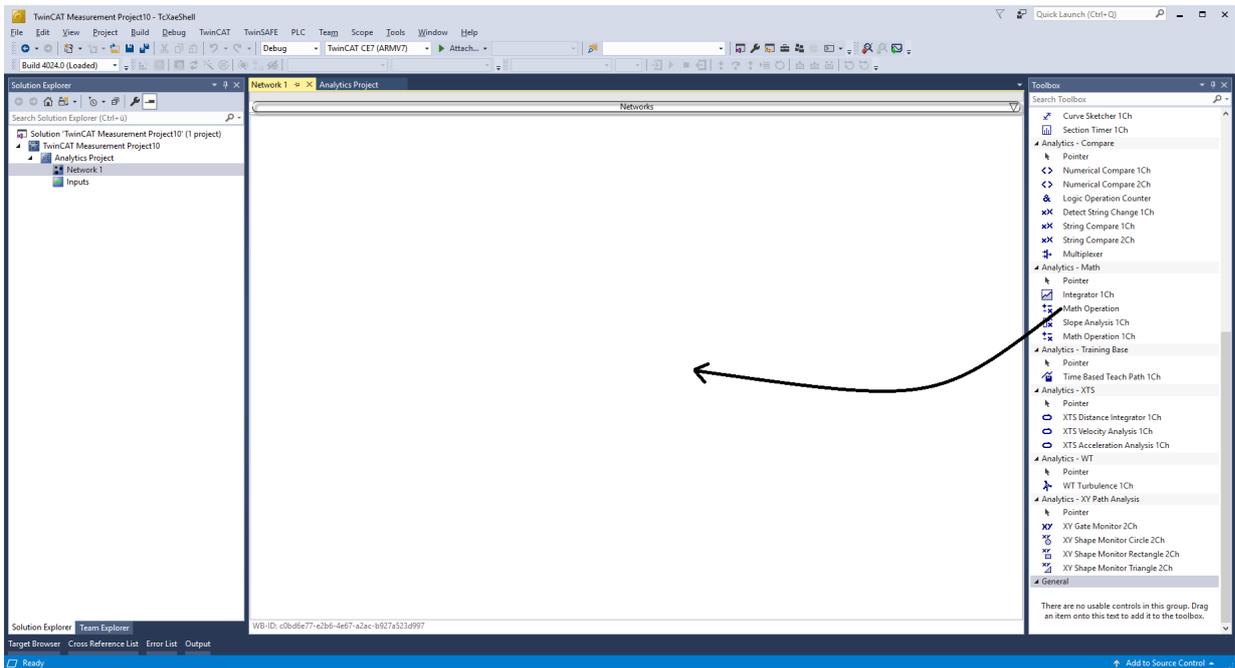
2. Wählen Sie aus **TwinCAT Measurement** die **Analytics-Projektvorlage** aus.



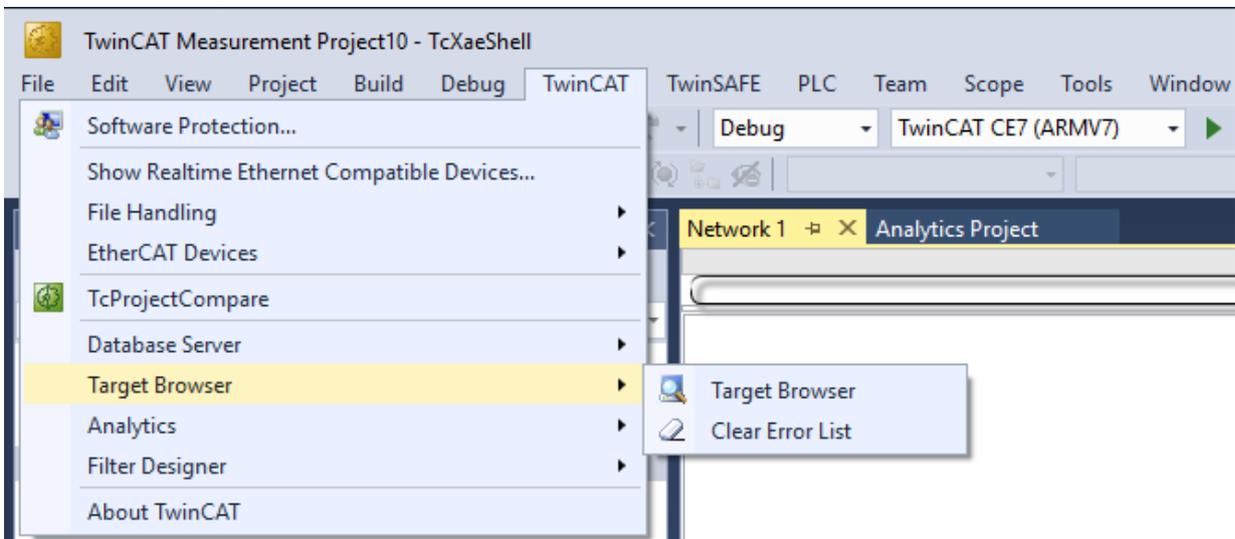
⇒ Das neue Projekt wird im Projektmappen-Explorer angezeigt. Nach einem Klick auf das Baumknotenenelement **Analytics Project** öffnet sich ein Startfenster, in dem Sie Ihre erste Aktion auswählen können. Von hier aus können Sie ein Netzwerk hinzufügen, die **Toolbox** öffnen, den **Target Browser** öffnen oder den **Analytics Storage Provider Recorder** öffnen. In den folgenden Schritten werden Sie all diese Aktionen durchführen.



- Es ist sinnvoll, als Erstes die **Toolbox** von Visual Studio® zu öffnen. Dort finden Sie alle unterstützten Algorithmen von TwinCAT Analytics. Die Algorithmen müssen in Netzwerke gruppiert und organisiert werden. Rechtsklicken Sie auf **Analytics Project**, um ein neues Netzwerk hinzuzufügen, oder fügen Sie mit Hilfe der Startseite ein Netzwerk hinzu. Das erste Netzwerk wird immer standardmäßig generiert.

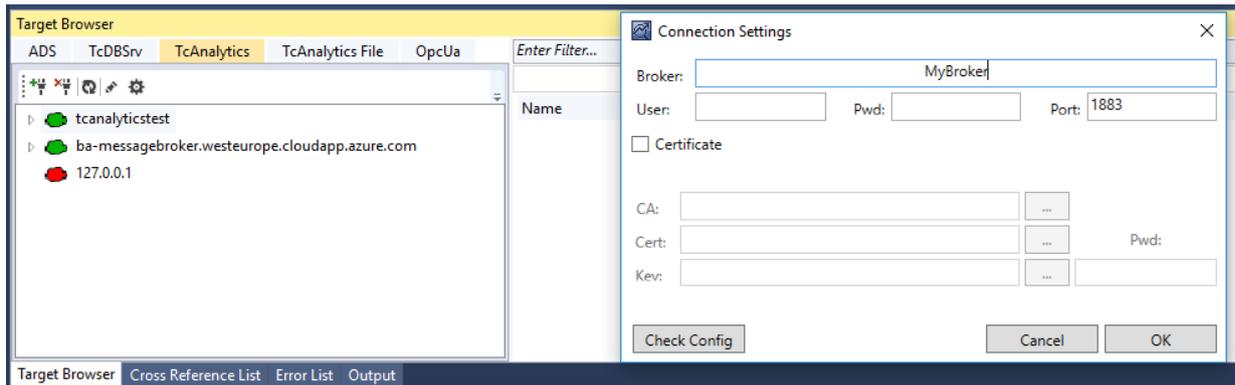


- Wenn Sie auf das Netzwerk klicken, wird ein Editor geöffnet. Nun können Sie den gewünschten Algorithmus per Drag-and-Drop in die Editor-Oberfläche ziehen.
- Nach der Auswahl des Algorithmus müssen Sie Eingangsvariablen mit den Modulen (Algorithmus) verbinden. Öffnen Sie dazu den **Target Browser**.  
**TwinCAT > Target Browser > Target Browser**



- Wählen Sie nun den Karteireiter **TcAnalytics** oder **TcAnalyticsFile** im Target Browser aus. Mit dem Karteireiter **TcAnalytics** (MQTT) geht es weiter.

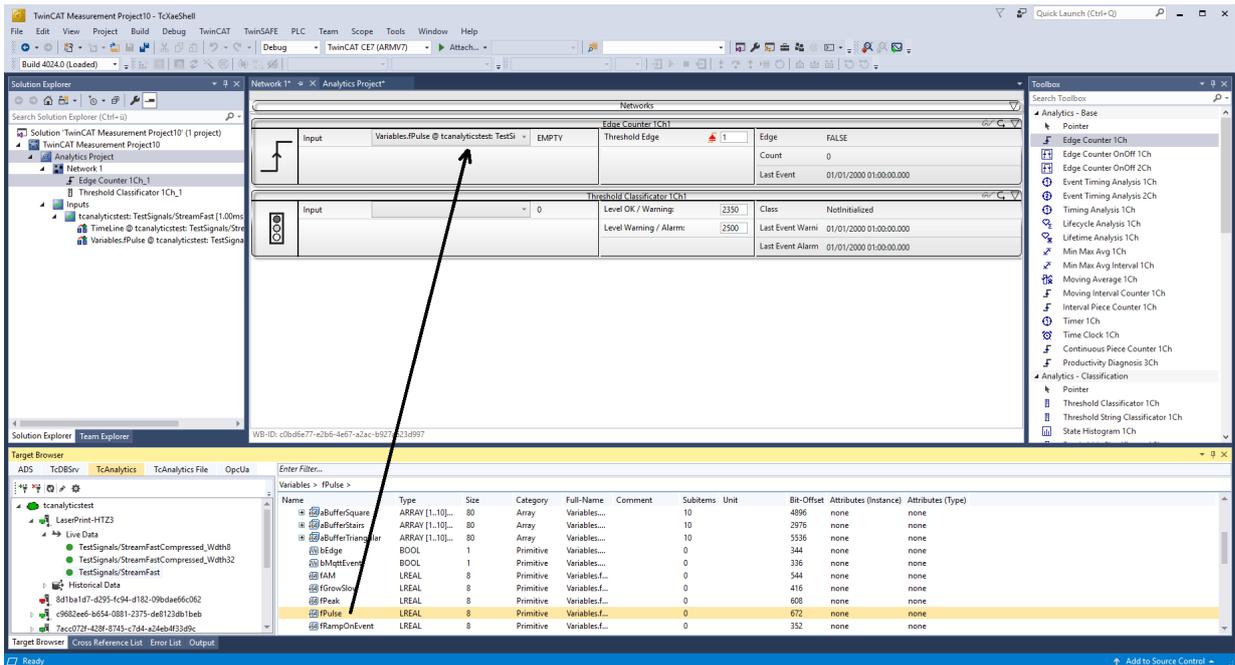
7. Klicken Sie auf das grün markierte Icon in der Symbolleiste dieser Analytics-Erweiterung. Es öffnet sich ein Fenster, in dem Sie die Konnektivitätsdaten Ihres Message-Brokers angeben können.



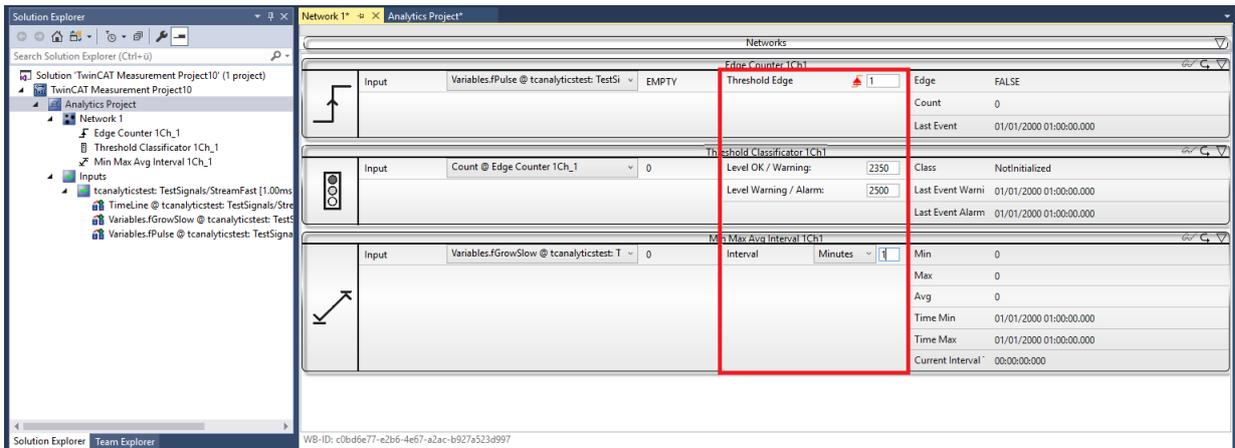
8. Wählen Sie Ihren MQTT-Analytics-Client aus (TwinCAT Analytics Logger, TwinCAT IoT Data Agent oder Beckhoff EK9160). Für jede Steuerung gibt es eine eindeutige ID. Diese ID wird im Target Browser angezeigt.
9. Über einen Klick auf das **Zahnrad-Icon**, gelangen Sie zur Seite Machine Administration. Hier können Sie einen System Alias-Namen vergeben, der im Target Browser anstelle der ID angezeigt wird.

Source	Customer	SystemID	System Alias	Online	Position	Topic Alias	Description
172.17.62.145							
TwinCAT Analytics Logger	MA Laser	3db95703-29fb-d99e-eb13-017b54677bb0	LaserPrintZG15	True			
TwinCAT Analytics Logger	MA Laser	3db95703-29fb-d99e-eb13-017b54677bb0	LaserPrintZG15	True	Laserstr. 13, 40597 Düsseldorf		MP4 Steel Laser Data
TwinCAT Analytics Logger	MA Laser	3db95703-29fb-d99e-eb13-017b54677bb0	LaserPrintZG15	True			
TwinCAT Analytics Logger	Fertig Motors	8d1ba1d7-d295-fc94-d182-09bdae66c062		False			
TwinCAT Analytics Logger	Fertig Motors	8d1ba1d7-d295-fc94-d182-09bdae66c062		False			
TwinCAT Analytics PLC DataLogger		7acc072f-428f-8745-c7d4-a24eb4f33d9c		False			
TwinCAT Analytics TEST Logger	Beckhoff	91c6eab3-1abb-5593-3651-1426874c91f		True	Hülshorstweg 20, 33415 Verl		
TwinCAT Analytics Storage Provider		c9682ee6-b654-0881-2375-de8123db1beb		True		RetroFit	
TwinCAT Analytics Storage Provider		7acc072f-428f-8745-c7d4-a24eb4f33d9c		True		AnalyticsSolution Results 1	
TwinCAT Analytics Storage Provider		7fb4f250-c130-7d7e-0a26-71ed8cee9340		True		CloudControl	
TwinCAT Analytics Storage Provider		3db95703-29fb-d99e-eb13-017b54677bb0	LaserPrintZG15	True			
TwinCAT Analytics Storage Provider		3db95703-29fb-d99e-eb13-017b54677bb0	LaserPrintZG15	True		TestSignals whole Image	
TwinCAT Analytics Storage Provider		c5eefcfd-4f14-5f45-dce4-7524715a9ae3		True		DataAgent Test	
TwinCAT Analytics Storage Provider		3f8a342a-6fac-3e76-6172-e7b5f62c0eb0		True		BigI40 FavValues	
TwinCAT Analytics Storage Provider		a313c550-7537-0617-827d-c6930e90d931		True		EK Test2	
TwinCAT Analytics Storage Provider		d180dde-afea-78d2-9ac1-65101d008687		True		NewMachineApp	
TwinCAT Analytics Storage Provider		3db95703-29fb-d99e-eb13-017b54677bb0	LaserPrintZG15	True		LongTerm	
TwinCAT Analytics Storage Provider		3db95703-29fb-d99e-eb13-017b54677bb0	LaserPrintZG15	True		MyFavoriteData	
TwinCAT Analytics Storage Provider		56cfbec6-3ab5-c1cc-1a1d-e6f4da86adf0		True		EdgeComputingTc2	
TwinCAT Analytics Storage Provider		56cfbec6-3ab5-c1cc-1a1d-e6f4da86adf0		True		EdgeComputingTc3	

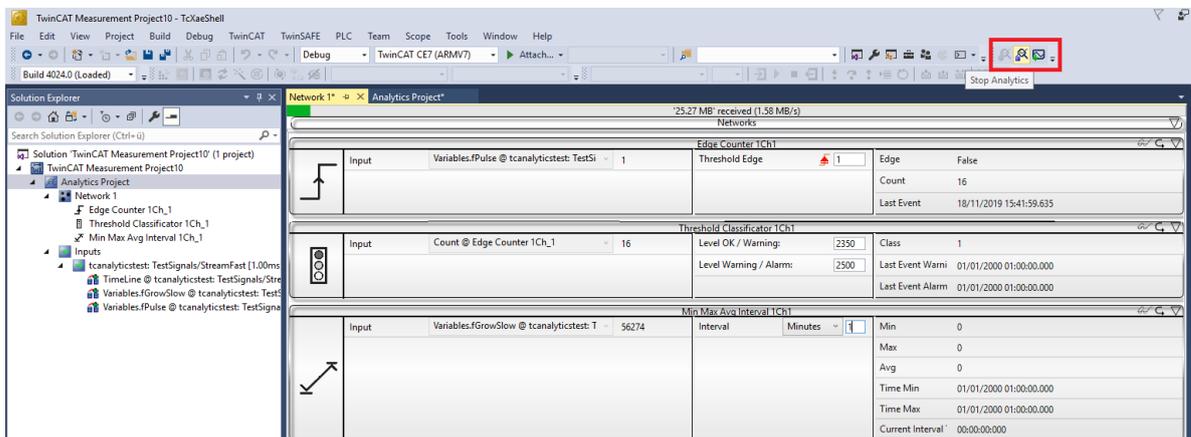
10. Im nächsten Schritt können Sie zwischen Live-Daten und historischen Daten für jeden MQTT Analytics-Client wählen. Die historischen Daten werden in diesem Fall von dem TwinCAT Analytics Storage Provider bereitgestellt.



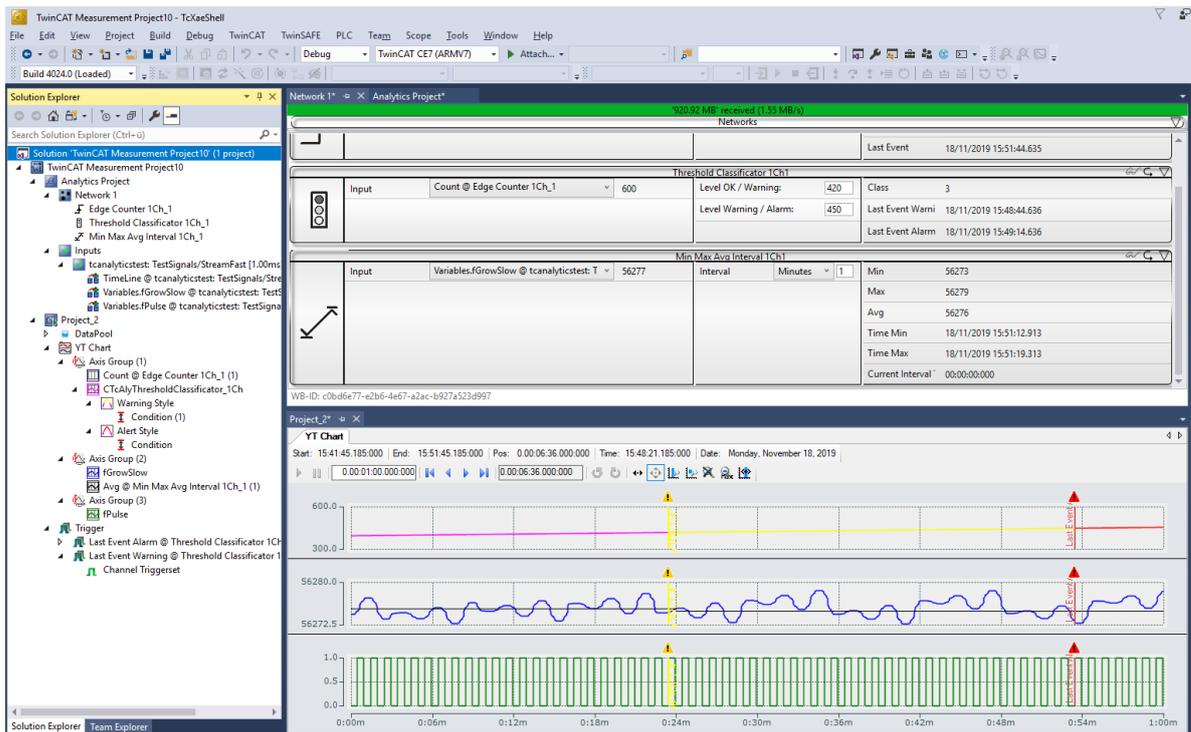
11. Sie können die Variablen per Drag-and-Drop in die Eingänge des spezifischen Algorithmus ziehen. Bei den meisten Algorithmen können Bedingungen wie Schwellen, Zeitintervalle, logische Operatoren usw. festgelegt werden. Diese Einstellungen werden in der Mitte jedes Moduls vorgenommen.



⇒ Schließlich ist Ihr erstes Analytics Project fertiggestellt. Um die Analyse zu starten, klicken Sie auf **Start Analytics**. Um die Analyse zu stoppen, klicken Sie auf **Stop Analytics**.



- ⇒ Vor dem Start der Analyse oder auch während der Laufzeit können Sie auf die Schaltfläche **Add Reference Scope** klicken. Dadurch wird automatisch eine zu Ihrem Analytics-Projekt passende Scope-Konfiguration erstellt.

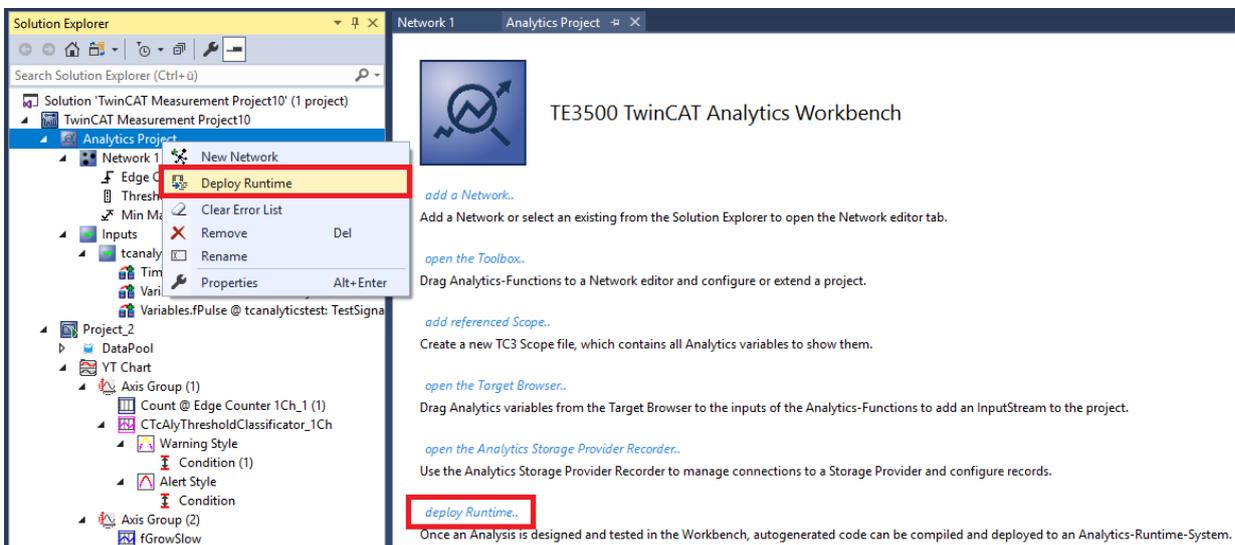


- ⇒ Die Ergebnisse der Analyse können in den Scope View-Grafiken per Drag-and-Drop angezeigt werden. Beispielsweise kann ein Mittelwert als neuer Kanal in der Ansicht angezeigt werden. Zeitstempel als Marker auf den X-Achsen zeigen signifikante Werte.

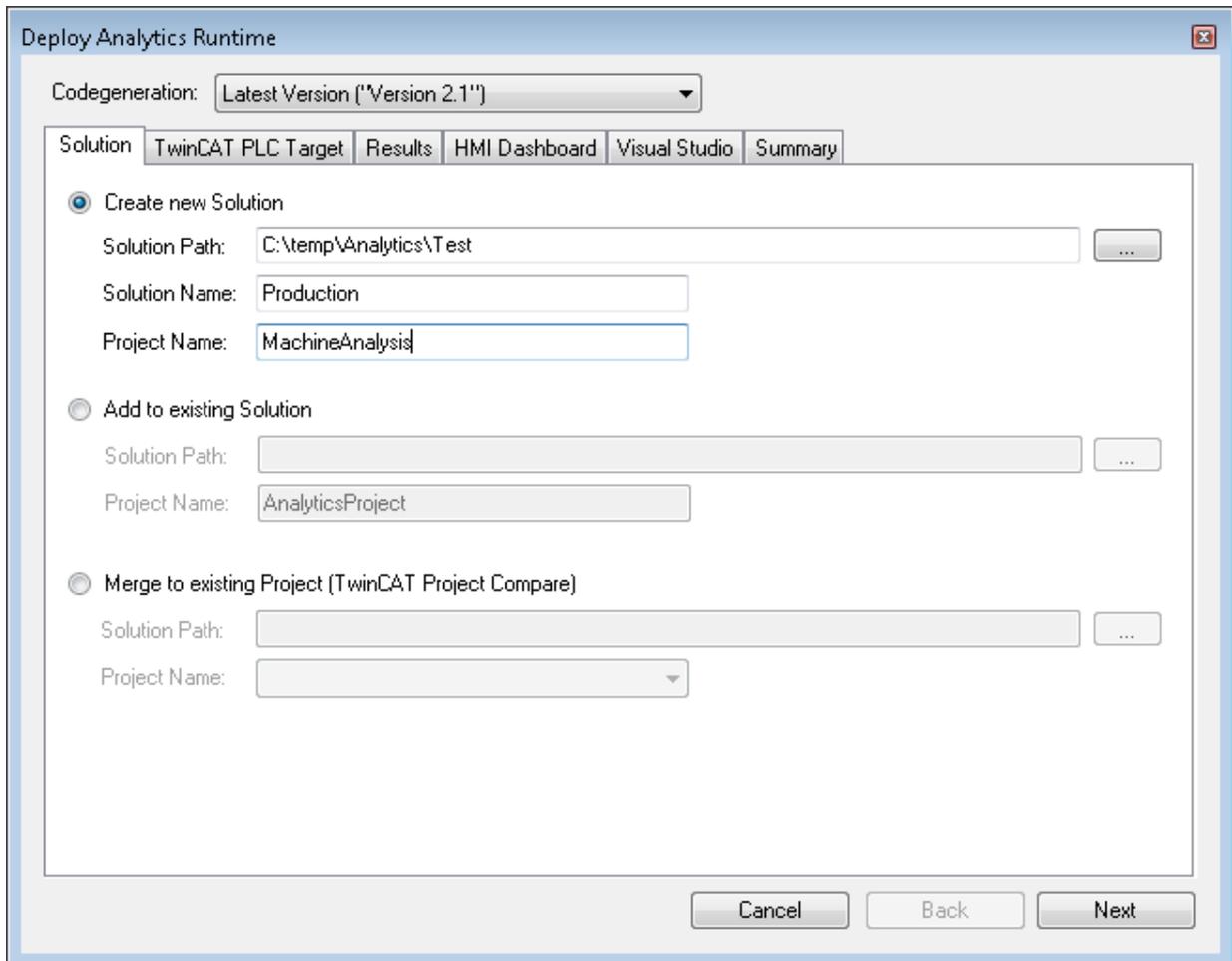
## 4.5 24-stündige Anwendung von Analytics

Der letzte große Schritt im Workflow von TwinCAT Analytics ist die kontinuierliche 24-stündige Maschinenanalyse. Sie läuft parallel zu den Maschinenanwendungen im Feld. Um dies ganz einfach zu gestalten, kann die TwinCAT Analytics Workbench automatisch einen SPS-Code und ein HTML5-basiertes Dashboard Ihrer Analytics-Konfiguration generieren. Beide können in eine TwinCAT Analytics Runtime (TC3 PLC und HMI Server) heruntergeladen werden und bieten dieselben Analyseergebnisse wie das Konfigurator-Tool in der Engineering-Umgebung.

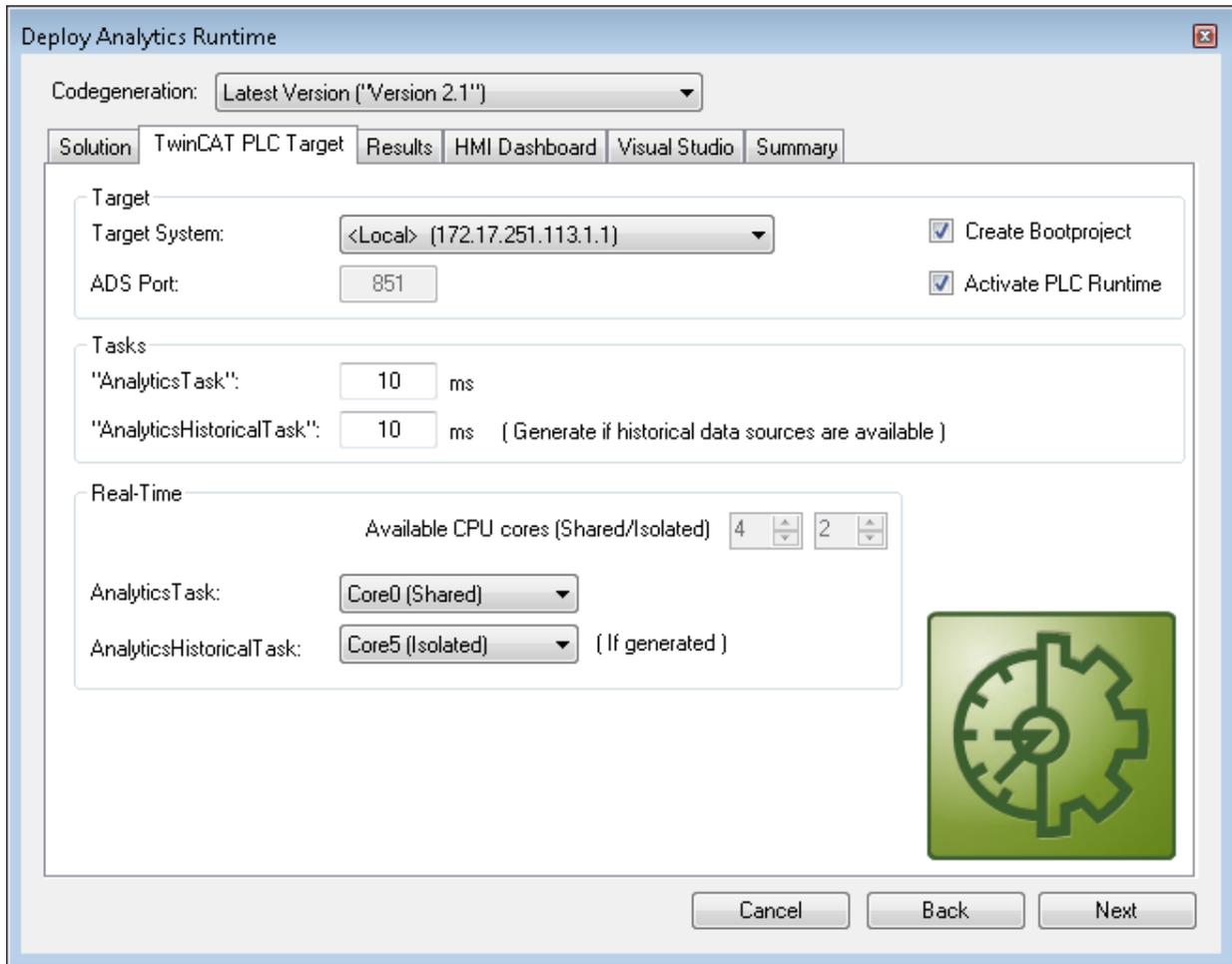
- ✓ Speichern Sie zunächst Ihre Konfiguration und öffnen Sie den Analytics Deploy Runtime-Assistenten. Dies kann über das Kontextmenü im Analytics Project-Tree Item oder über die Startseite erfolgen.



1. Wenn der Assistent geöffnet ist, können Sie sich durch einige Karteireiter klicken. Der erste heißt Solution. Hier können Sie entscheiden, wie Ihr Analytics-Projekt im SPS-Code verwendet werden soll:  
 Als...  
 vollständig neue Lösung.  
 Teil einer vorhandenen Lösung.  
 Aktualisierung einer vorhandenen Analytics-Lösung.



2. Im Karteireiter **TwinCAT PLC Target** können Sie das ADS-Zielsystem wählen, das die TwinCAT Analytics Runtime (TF3550) ausführt. Das erzeugte Projekt ist sofort ausführbar. Hierzu können Sie die Option **Activate PLC Runtime** setzen. Zudem kann ausgewählt werden, dass direkt ein Bootprojekt erstellt wird.



3. Insbesondere bei virtuellen Maschinen ist es wichtig, das Projekt auf isolierten Kernen auszuführen, was ebenfalls eine Option in diesem Karteireiter ist. Der nächste Karteireiter **Results** wird nur benötigt, wenn Sie in den Algorithmeigenschaften die Option **Stream Results** ausgewählt haben. Wenn Sie Ergebnisse senden möchten, können Sie hier entscheiden, in welcher Weise (lokal in einer Datei/durch MQTT) und welchem Format (binär/JSON) dies geschehen soll. Auch dies wird automatisch generiert und umgehend nach Aktivierung ausgeführt.

Deploy Analytics Runtime

Codegeneration: Latest Version ("Version 2.1")

Solution | TwinCAT PLC Target | **Results** | HMI Dashboard | Visual Studio | Summary

Create no Results

Stream Results to MQTT Broker

Topic: Analytics/Analysis/ResultStream

MQTT Connection Settings Json Format

Write Results to Analytics File

File Path:  ...

Max File Size: 256 Sample buffer count

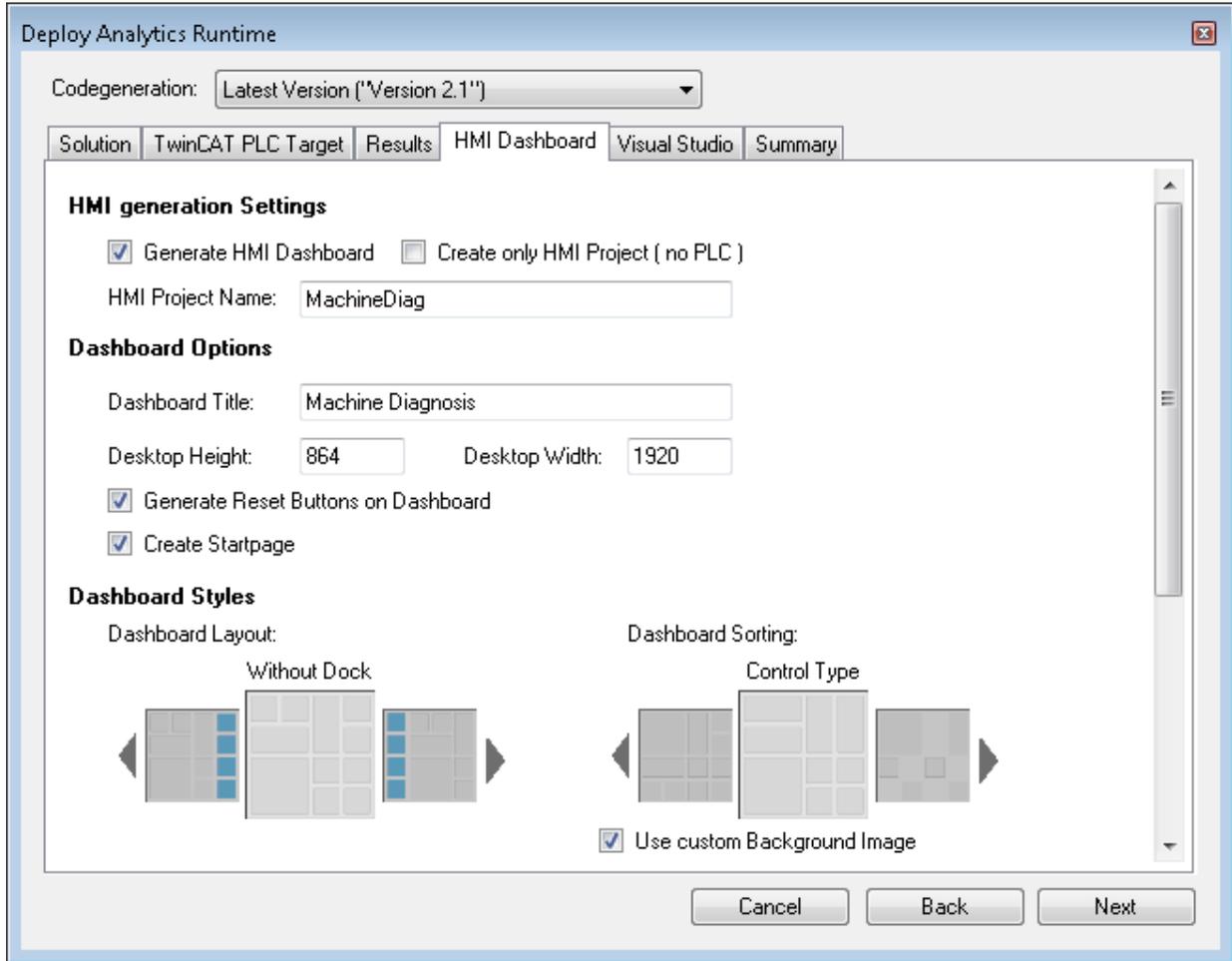
Select Result Items

CycleTime: User specified cycle time 5000 ms

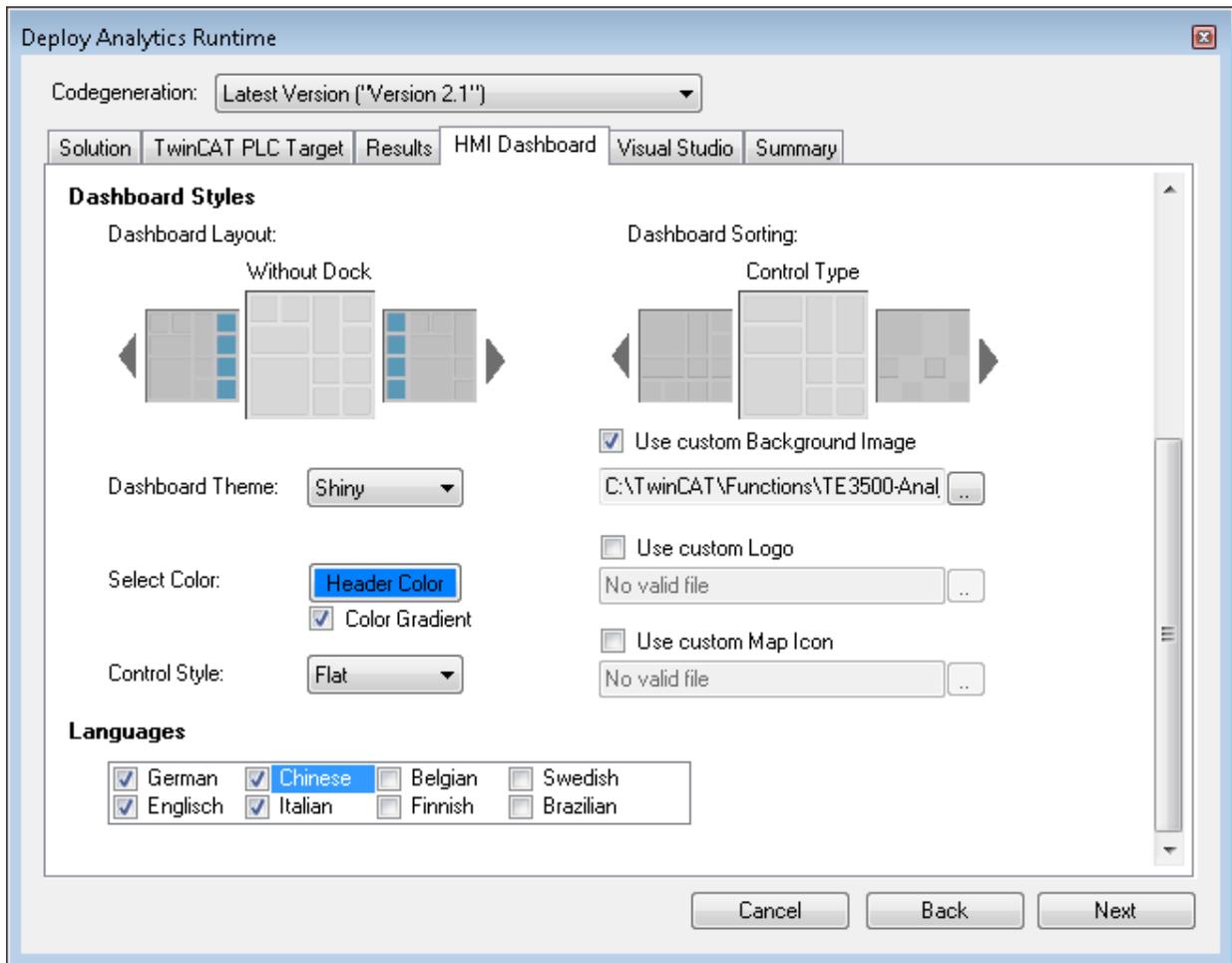
Cancel Back Next

Ein Downsampling der Ergebnisse ist durch die Festlegung einer Zykluszeit möglich. Der nächste Karteneiter ist für das **HMI-Dashboard** vorgesehen. Voraussetzung für die automatische Generierung

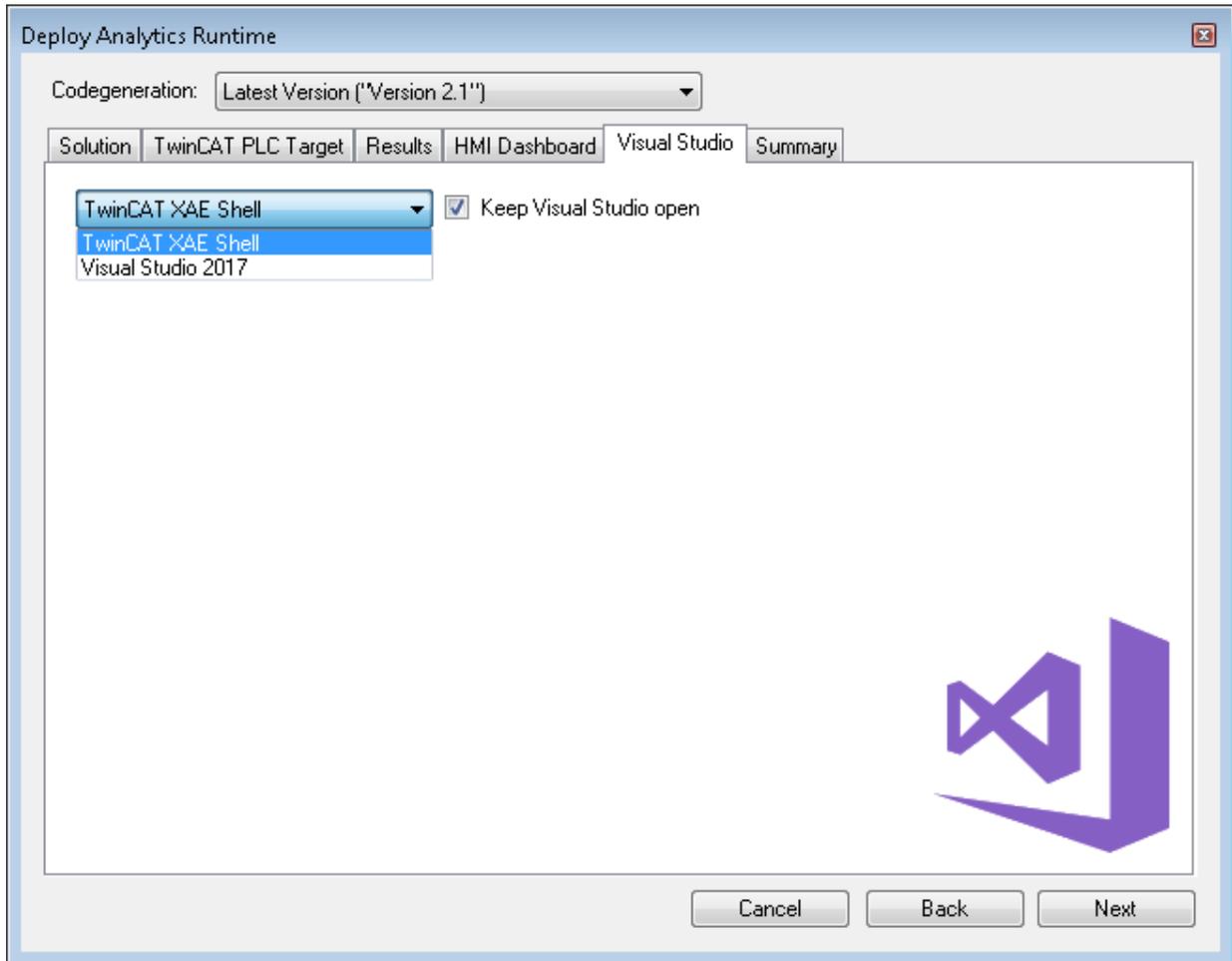
des Dashboards ist die Auswahl von HMI Controls für die entsprechenden Algorithmen, deren Ergebnisse im Dashboard angezeigt werden sollen.



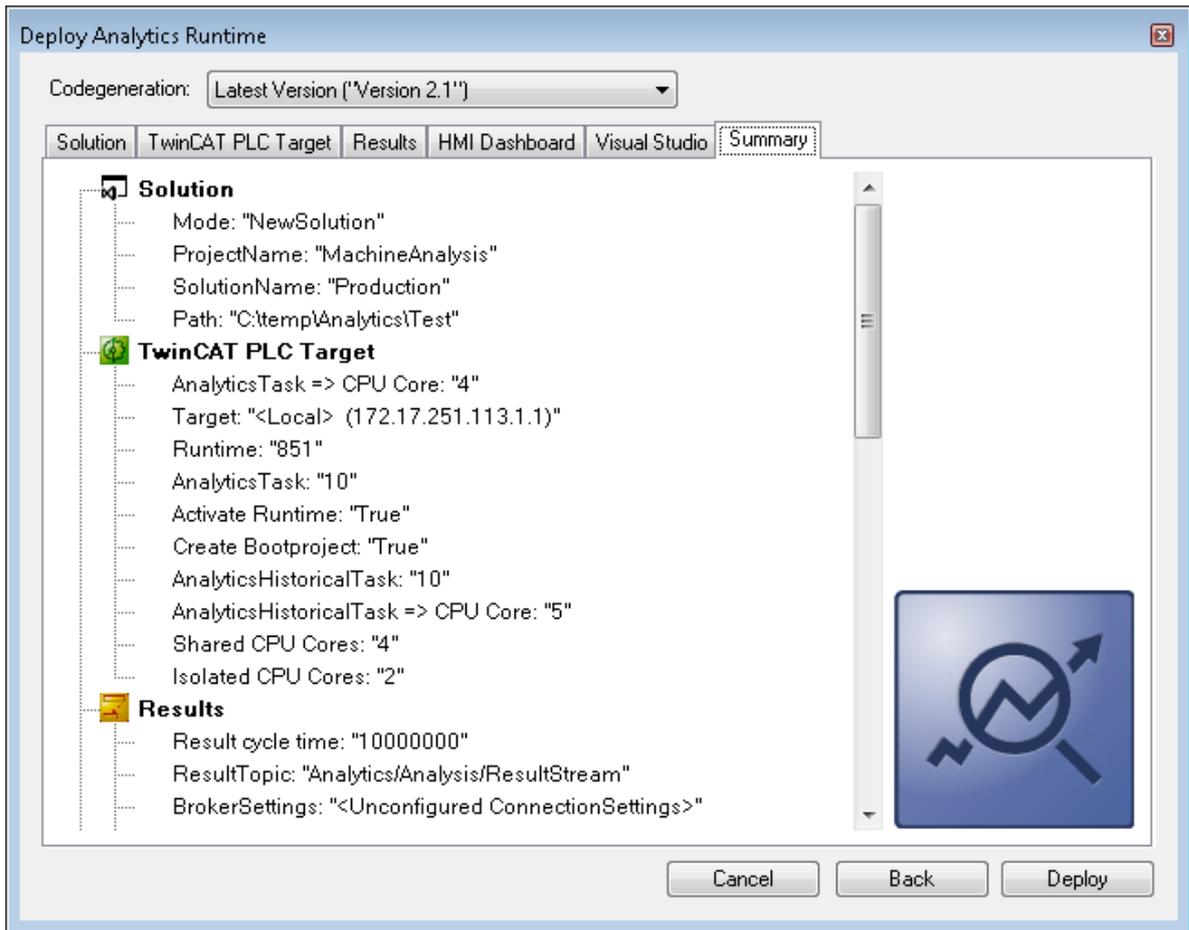
- Sie können verschiedene Optionen für Ihr Analytics Dashboard auswählen, wie eine Startseite mit einer Karte, Layouts, Sortieralgorithmen, eigene Farben und Logos. Wenn Sie mehrere Sprachen für die Analytics Controls auswählen, wird auch ein Menü zur Sprachumschaltung generiert.



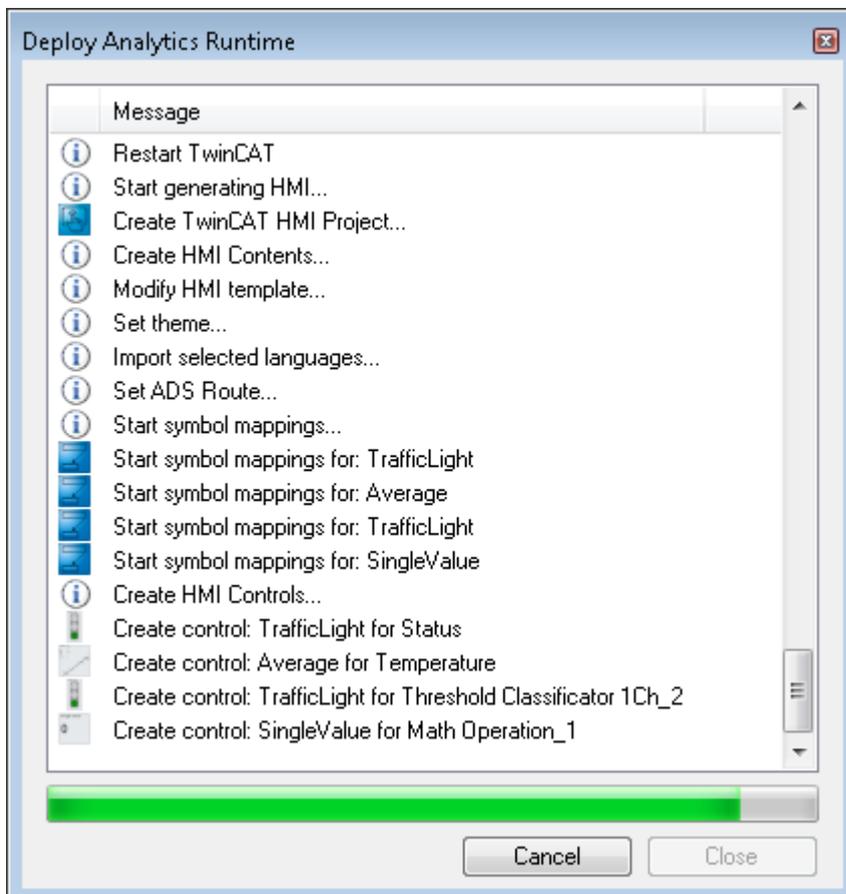
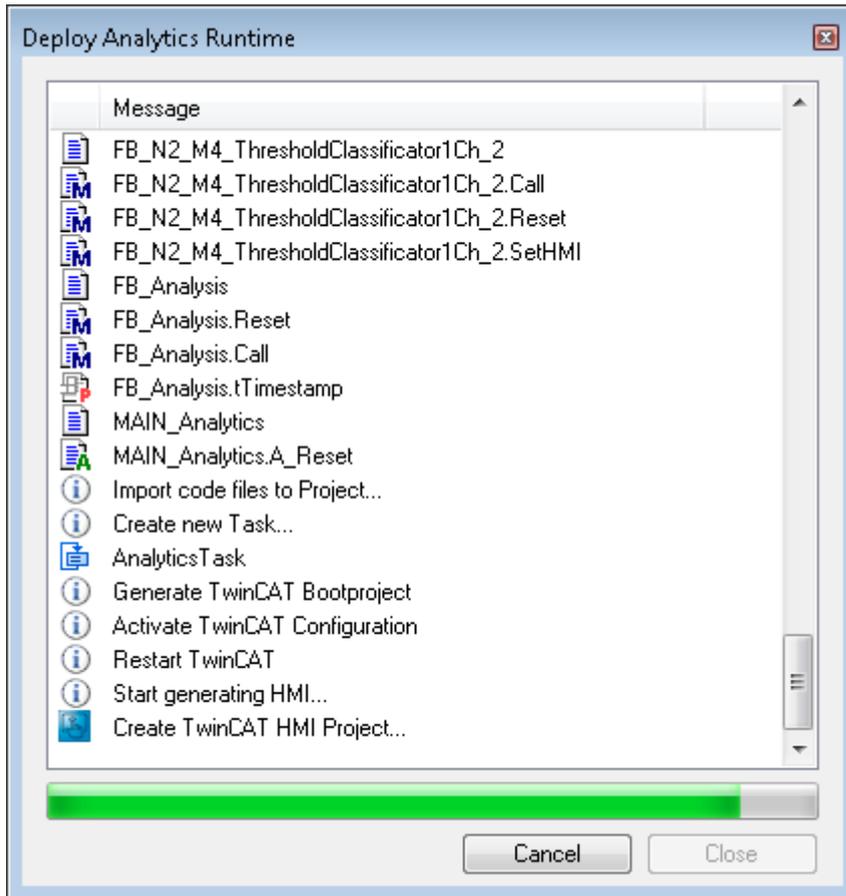
5. Wählen Sie eine der installierten Versionen von Visual Studio® aus, und, ob die Instanz sichtbar starten soll oder ob sie nur eingerichtet und im Hintergrund aktiviert werden soll.



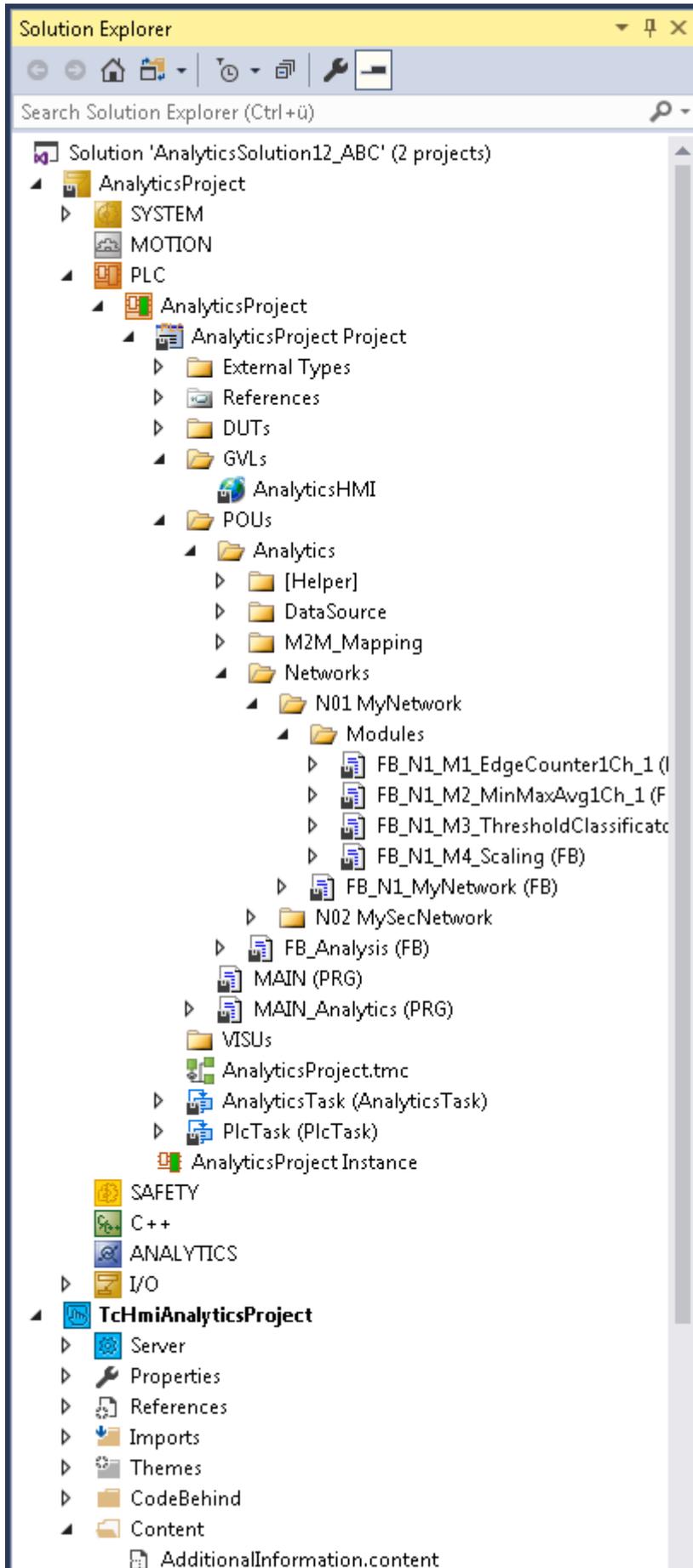
⇒ Zuletzt finden Sie eine Übersicht.



- Nun können Sie auf die Schaltfläche **Deploy** klicken, um den Generierungsprozess zu starten. Das SPS-Projekt und das HMI-Dashboard werden nun generiert.



⇒ Nach der Meldung „Deploy Runtime succeeded“ finden Sie eine neue Visual Studio®/XAE Shell-Instanz auf Ihrem Desktop. Die neue Projektmappe und beide Projekte werden erstellt.



## 5 Technische Einführung

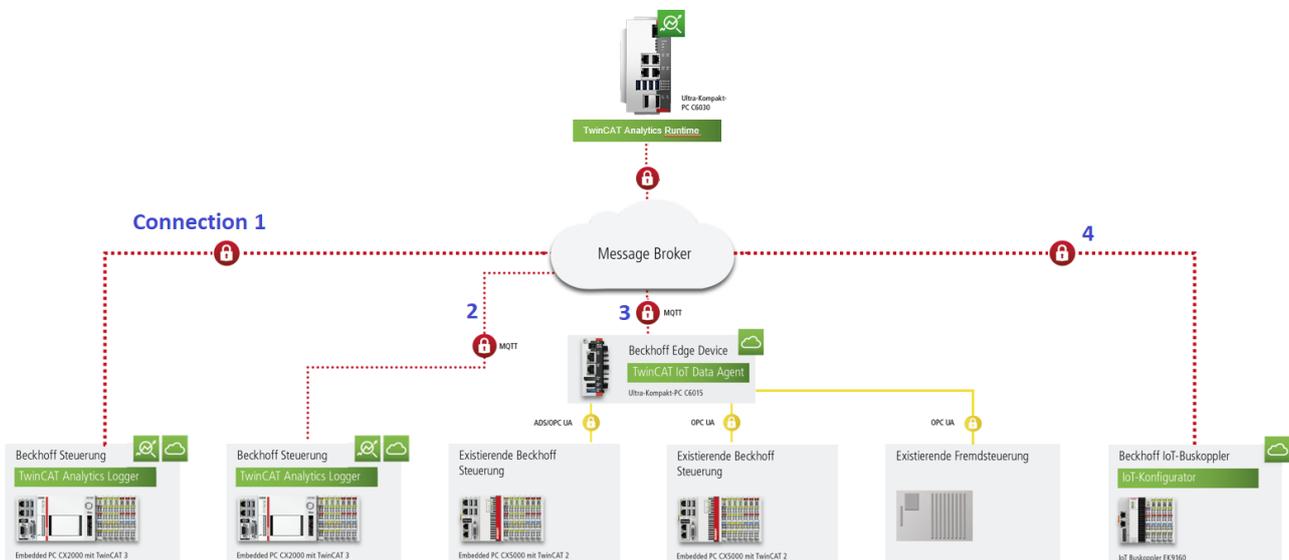
Hinsichtlich des Workflows von TwinCAT Analytics ist die Analytics Runtime in der Lage, eine kontinuierliche Analyse von Daten von verschiedenen Remote-Steuerungen zu erstellen. Die Analytics Runtime basiert auf einer Standard TwinCAT PLC Runtime. Der Anwendungscode kann durch die Analytics Workbench automatisch generiert werden. So hat der Benutzer keinen zusätzlichen Engineering-Aufwand, aber es steht ihm frei, Codeänderungen vorzunehmen oder eigenen Analysecode hinzuzufügen. Auch die Nutzung anderer Standard SPS-Bibliotheken von Beckhoff, wie Tc3\_Database oder Tc3\_ConditionMonitoring, ist möglich.

### Verbindungshandhabung mit Remote-Geräten

Die Anzahl der zu analysierenden Geräte/Steuerungen ist begrenzt. Die Analytics Runtime selbst ermöglicht vier Verbindungen zu Steuerungen ohne zusätzliche Pakete.

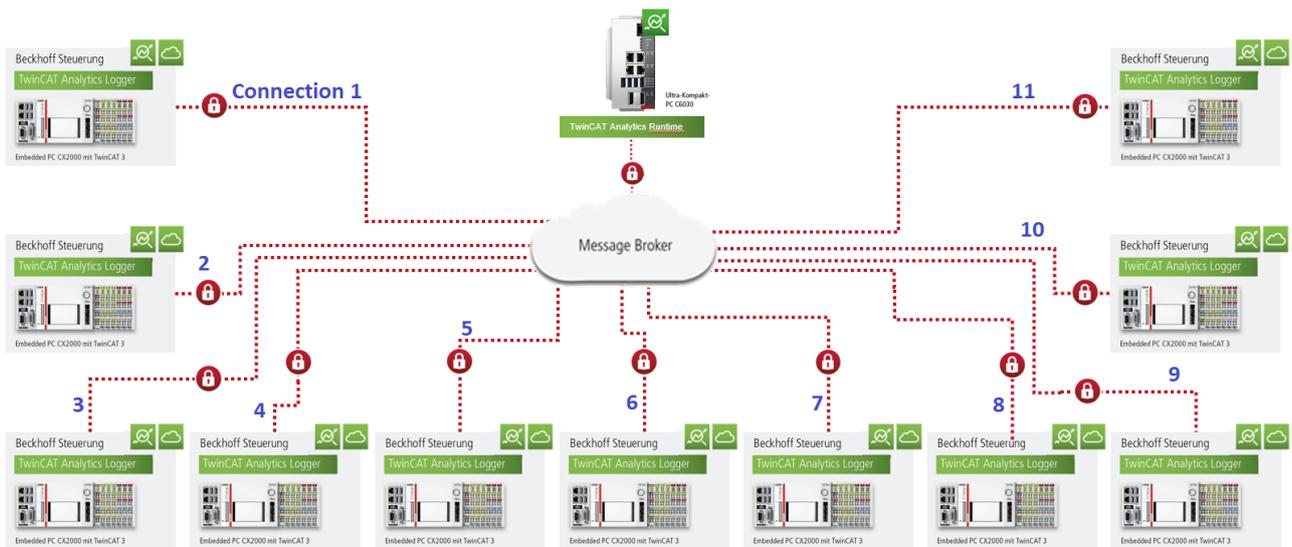
Derzeit wird **eine** Verbindung für die folgenden Steuerungen gezählt:

- Steuerung mit TF3500 Analytics Logger.
- Steuerung mit TF6720 IoT Data Agent (kann viele unterlagerte Steuerungen haben).
- EK9160 IoT-Buskoppler.



Wenn Sie mehr als vier Verbindungen in Ihrer Anwendung benötigen, können Sie Controller Packs hinzufügen. Mögliche Packs sind hier aufgeführt.

Das Bild unten zeigt eine Anwendungsarchitektur mit 11 verbundenen Steuerungen. In diesem Fall müssen Sie zusätzlich zu der TwinCAT Analytics Runtime-Lizenz eine Lizenz für Controller Pack 8 auf dem Analytics Runtime Gerät/VM bestellen.



**TwinCAT HMI**

Basierend auf dem Analytics-SPS-Code kann der Techniker ein eigenes HTML5-Dashboard mit TE2000 gestalten, das in das Setup der TE3500 Analytics Workbench integriert wird. Die Analytics Runtime enthält den HMI Server. Der HMI Server stellt die HTML5-Webseiten für mindestens vier Clients gleichzeitig bereit. Der Server selbst bietet eine Client-Verbindung und die Analytics Runtime umfasst die Lizenz für das Client Pack 3. Dies bietet Ihnen die Möglichkeit von 4 Verbindungen.



## 6 Konfiguration

Die Konfiguration der Analytics Runtime erfolgt in dem Engineering-Tool TE3500 Analytics Workbench. Die notwendigen Schritte werden hier erneut erläutert. Falls Sie weitere Informationen über die Konfiguration in TE3500 benötigen, wechseln Sie bitte zur Dokumentation von TE3500.

### 6.1 Runtime-Deployment

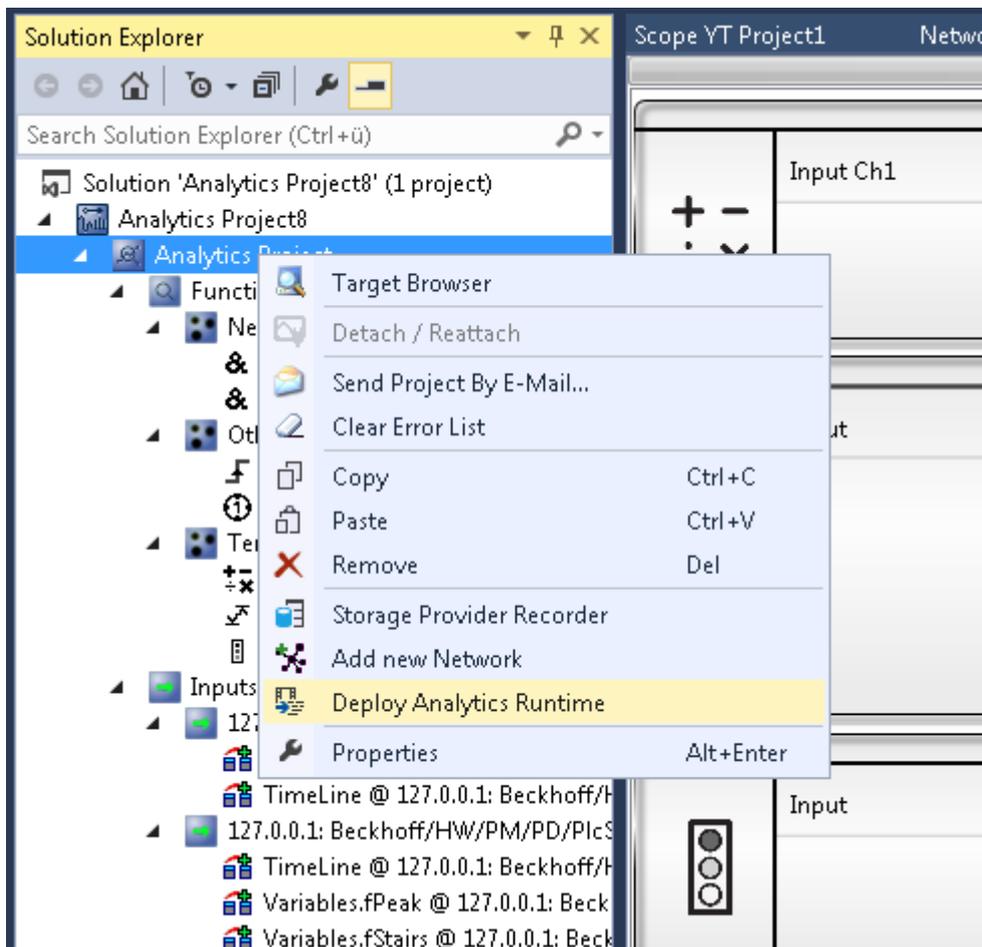
SPS-Code kann mit allen Modulen und Parametern generiert werden, die im TwinCAT Analytics Workbench Configurator konfiguriert sind. Dieser Code kann in eine TwinCAT Analytics Runtime heruntergeladen werden, um eine 24/7-Datenanalyse zu realisieren.

#### HINWEIS

##### Kompatibilität von automatisch generiertem SPS-Code

Der automatisch generierte SPS-Code basiert auf der TwinCAT Analytics Library. Schnittstellen der Basismodelle der Bibliothek sind mit früheren Versionen nach Freigabe der Bibliothek kompatibel. Der automatisch generierte Code selbst ist nur ein Beispielcode! Diese Codegenerierung kann sich ggf. von Version zu Version ändern. Soweit möglich, wird dies durch die Codegenerierungsversion gelöst.

1. Nach der Konfiguration können Sie, wie im Bild unten gezeigt, auf den Befehl **Deploy Analytics Runtime** im Kontextmenü klicken.



- Der Deploy-Assistent startet und es ist möglich, die gesamte benötigte Konfiguration für den Einsatz schrittweise einzurichten.

Deploy Analytics Runtime

Codegeneration: Latest Version ("Version 4.0")

Solution | TwinCAT PLC Target | Results | HMI Dashboard | Visual Studio | Summary

Create new Solution

Solution Path: C:\Users\Beckhoff\Documents

Solution Name: AnalyticsSolution

Project Name: AnalyticsProject PLC Name: AnalysisPLC

Add to existing Solution

Solution Path:

New TwinCAT Project  New PLC Project  Include to PLC Project

Project Name: AnalyticsProject PLC Name: AnalysisPLC

Merge to existing Project (TwinCAT Project Compare)

Solution Path:

Project Name: PLC Name:

Cancel Back Next

- Auf dem ersten Karteireiter **Solution** kann ausgewählt werden, ob eine neue Projektmappe erzeugt wird, ob in eine bereits bestehende Projektmappe der Analyse SPS Code als neues Projekt, neue PLC oder in eine bestehende PLC integriert bzw. hinzugefügt wird, oder ob das neue Projekt in eine vorhandene Projektmappe mit Hilfe des TwinCAT Project Compare Tool eingefügt wird.

Deploy Analytics Runtime

Codegeneration: Latest Version ("Version 4.0")

Solution | TwinCAT PLC Target | Results | HMI Dashboard | Visual Studio | Summary

Target

Target System: DESKTOP-P3914EA (172.17.251.193.1.1)  Create Bootproject

ADS Port: 851  Enable UTF8 Encoding  Activate PLC Runtime

Tasks

"AnalyticsTask": 10 ms

"AnalyticsHistoricalTask": 10 ms (Generate if historical data sources are available)

Real-Time

Available CPU cores (Shared/Isolated) 3 1

AnalyticsTask: Core3 (Isolated)

AnalyticsHistoricalTask: Core3 (Isolated) ( If generated )

Cancel Back Next

4. Auf der zweiten Registerkarte **TwinCAT PLC Target** können alle SPS-spezifischen Parameter wie Target System, Task-Zykluszeit, oder auch die Taskzuordnung zu entsprechenden CPU-Kernen, festgelegt werden.

Deploy Analytics Runtime ✕

Codegeneration: Latest Version ("Version 4.0")

Solution | TwinCAT PLC Target | Results | HMI Dashboard | Visual Studio | Summary

Create no Results

Stream Results to MQTT Broker

Topic:

messagebroker.beckhoff-cloud.com

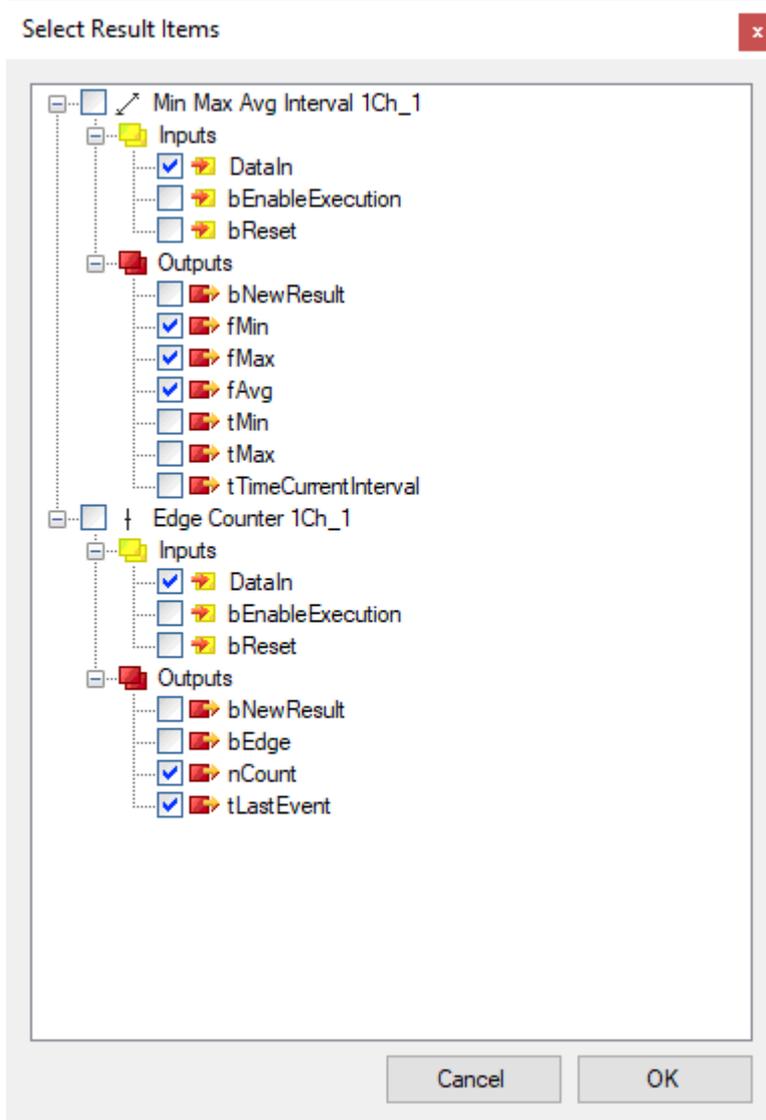
Write Results to Analytics File

File Path:

Max File Size:    Sample buffer count

CycleTime:

5. Wenn Sie die Eigenschaft „PLC Result“ einiger Funktionen im Konfigurator gesetzt haben, ist der Karteireiter **Result** im Deploy-Assistenten freigeschaltet. Dort kann eingerichtet werden, wo die Ergebnisse gestreamt oder gespeichert werden.



6. Durch Anklicken von **Select Result Items** ist es möglich, nur die gewünschten Werte auszuwählen.

Deploy Analytics Runtime

Codegeneration: Latest Version ("Version 4.0")

Solution | TwinCAT PLC Target | Results | HMI Dashboard | Visual Studio | Summary

**HMI generation Settings**

Generate HMI Dashboard     Create only HMI Project (no PLC)

HMI Project Name: AnalyticsHMIProject

**Dashboard Options**

Dashboard Title: Analytics Dashboard

Desktop Height: 864    Desktop Width: 1920

Create Startpage     Show current time

**Dashboard Styles**

Dashboard Layout:    Dashboard Sorting:

Without Dock    Space Saving

Cancel    Back    Next

7. Wenn Sie ein oder mehreren Funktionen HMI Controls zugewiesen haben, ist der Karteireiter **HMI Dashboard** im Deploy-Assistenten freigeschaltet. Hier können verschiedene Einstellungen vorgenommen werden um, um ein maßgeschneidertes Dashboard zu generieren, das zu Ihren Bedürfnissen passt.

Deploy Analytics Runtime

Codegeneration: Latest Version ("Version 4.0")

Solution | TwinCAT PLC Target | Results | HMI Dashboard | Visual Studio | Summary

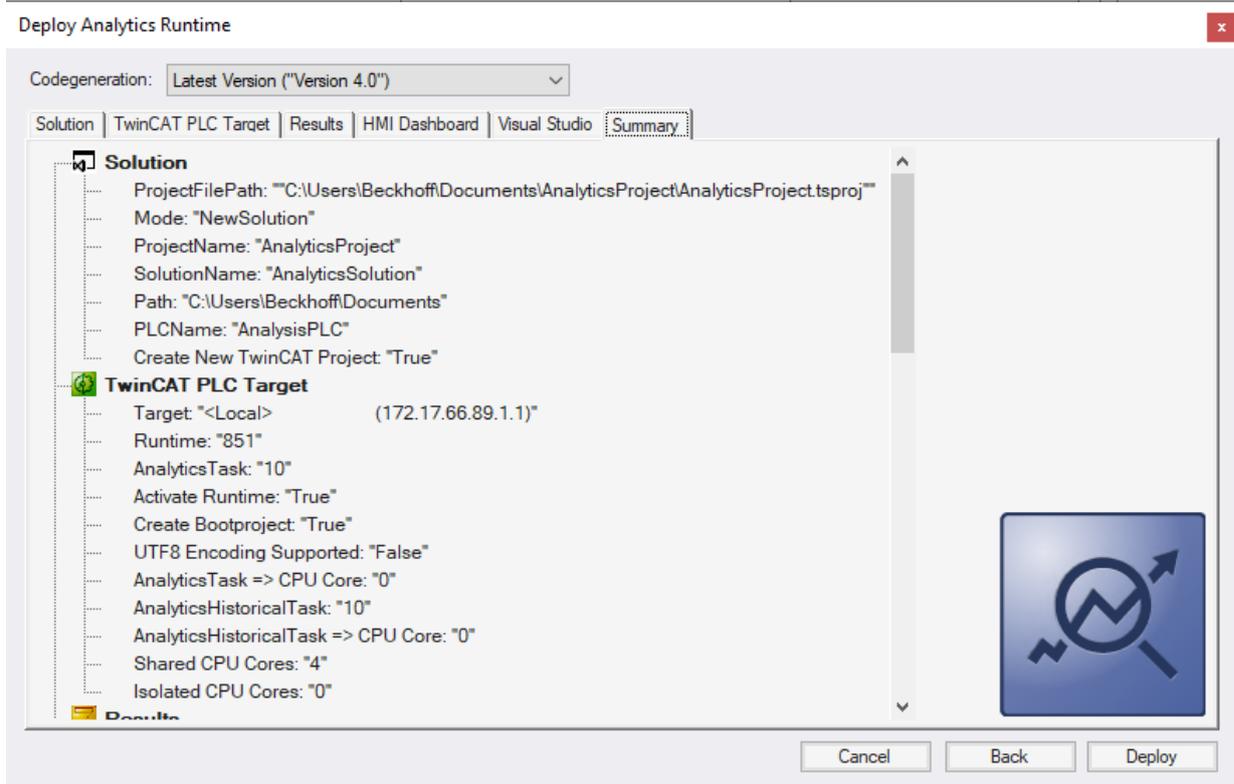
Keep Visual Studio open     Show Visual Studio during generating code

Target VS Version:  
TwinCAT XAE Shell

Create HMI in another Visual Studio  
TwinCAT XAE Shell

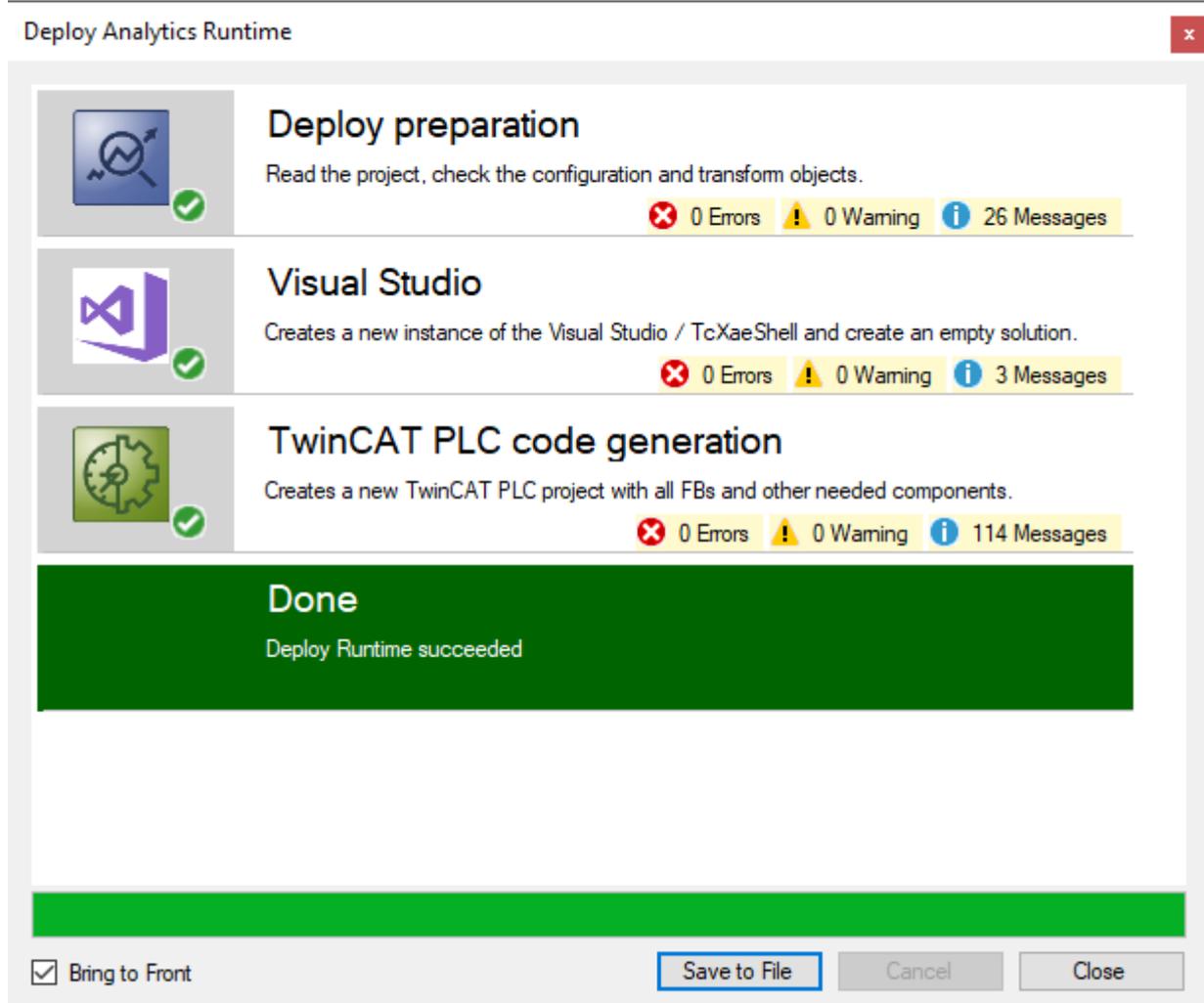
Cancel    Back    Next

8. Auf dem nächsten Karteireiter **Visual Studio** kann ausgewählt werden, welche Visual Studio®-Version oder TwinCAT XAE Shell für die Generierung verwendet werden soll, falls mehrere installiert sind.



9. Der letzte Karteireiter zeigt Ihnen alle Einstellungen, die Sie für die Generierung vorgenommen haben. Nun können Sie den Generierungsprozess starten, indem Sie auf **Deploy** klicken.

⇒ Im Übersichtsfenster wird jeder Schritt während des Generierungsprozesses, übersichtlich und in Kategorien unterteilt, angezeigt.



### 6.1.1 Algorithmeigenschaften

Jeder Algorithmus des Analytics-Configurators stellt einige Eigenschaften bereit. Die Abschnitte HMI und PLC sind für die automatische Codegenerierung notwendig.

Properties

Edge Counter 1Ch\_1 TwinCAT.Measurement.AnalyticsFunctionNodeProper

Appearance

Color	Black
Comment	
SortPriority	100

Function

ClassId	02040101-0000-0000-f000-000000000064
ClassName	
FactoryName	TcAnalyticsKernel
Group	TcAnalyticsKernel Base
Title	Edge Counter 1Ch_1

HMI

Generate GVL	False
GlobalVariableType	InOutVariables

Network

NetworkID	1
-----------	---

PLC

Persistent Results	False
Stream Results	False

Appearance

Toolbox Properties

#### HMI

- **Generate GVL:** Ermöglichen Sie die Erzeugung einer globalen Variablenliste mit einer Sammlung von Variablen und entsprechendem Datentyp-Mapping für TwinCAT HMI.
- **GlobalVariableType:** Wählen Sie den Typ mit InOutVariables nur für die Eingänge und Ausgänge des Algorithmus oder KeyValuePairs für eine allgemeine Zuordnung zum STRING für Tabellen.

#### PLC

- **Persistent Results:** Aktivieren Sie dieses Flag, um die Ergebnisse des Algorithmus dauerhaft im Zielsystem der Analytics Runtime zu speichern.
- **Stream Results:** Aktivieren Sie dieses Flag, um die Ein- und Ausgänge des Algorithmus zu einem Ergebnis-Stream hinzuzufügen, der durch die Codegenerierung erzeugt wird.

## 6.1.2 SPS-Code

```

1 PROGRAM MAIN_Analytics_C1
2 VAR_INPUT
3   stReset: ST_AnalysisReset;
4 END_VAR
5 VAR_OUTPUT
6   bError: BOOL;
7   ipTcResult: I_TcMessage;
8 END_VAR
9
10 A_Reset();
11
12 fbResults.Call();
13
14 IF fbResults.bInitialized THEN
15   IF TestAndSet(Analytics.bLock_T1) THEN
16     IF Analytics.fbT1_DataSource.bNewData THEN
17       WHILE NOT Analytics.fbT1_DataSource.bEndOfData DO
18         Analytics.fbT1_DataSource.NextData();
19
20         fbAnalysis.Call();
21         IF fbAnalysis.bError THEN
22           EXIT;
23         ELSE
24           fbAnalysis.ResultStream(fbResults);
25         END_IF
26       END_WHILE
27     ELSE
28       fbAnalysis.Call();
29     END_IF
30
31     Analytics.bLock_T1:= FALSE;
32   END_IF
33
34 //Error Handling
35 bError:= fbAnalysis.bError;
36 ipTcResult:= fbAnalysis.ipTcResult;

```

### HINWEIS

#### Kompatibilität von automatisch generiertem SPS-Code

Der automatisch generierte SPS-Code basiert auf der TwinCAT Analytics Library. Schnittstellen der Basismodelle der Bibliothek sind mit früheren Versionen nach Freigabe der Bibliothek kompatibel. Der automatisch generierte Code selbst ist nur ein Beispielcode! Diese Codegenerierung kann sich ggf. von Version zu Version ändern.

Codegenerierung Versions Kompatibilität

	Version 2.1 (veraltet)	Version 3.0 (veraltet)	Version 4.0	Version 4.1	Version 5.0
Analytics Algorithmen	(X)	(X)	X	X	X
Filter Algorithmen	(X)	(X)	X	X	X
HMI Support	(X)	(X)	X	X	X
HMI mit erweitertem Inputstream Handling	-	(X)	X	X	X
Unterstützung von Array Eingänge	(X)	(X)	X	X	X
Unterstützung von Oversampling Eingängen	-	-	X	X	X
Netzwerktemplates	(X)	(X)	X	X	X
„Closed“ Netzwerktemplates	-	-	X	X	X
Netzwerk mit Eingängen, Ausgängen und Parametern	-	-	X	X	X
„Condition Monitoring“ Netzwerktemplates	-	-	X	X	X
Lambda Algorithmen	-	-	X	X	X
Parameter schreiben per HMI	-	-	-	X	X
„Power Monitoring“ Algorithmen	-	-	X	X	X
Scope Support	-	-	-	-	X

### 6.1.2.1 Codeversion 2.1

The Solution Explorer shows the following structure:

- Task:** AnalyticsTask
- StreamHelper:** 172.17.62.145\_Obj1 (StreamHelper)
- DataTypes:** Analytics
  - HMI
  - Results
    - ST\_Results (STRUCT)
    - E\_AnalysisComponents (ENUM)
    - ST\_AnalysisReset (STRUCT)
- HMI GVL:** AnalyticsHMI
- DataSource / M2M Mapping:**
  - DataSource
    - T01 DataSource
      - FB\_T1\_DataSource (FB)
      - I\_T1\_DataSource
    - M2M\_Mapping
      - FB\_ValueMapping\_M2M (FB)
      - I\_ValueMapping\_M2M
  - Networks
    - N01 Network1
      - Modules
        - FB\_N1\_M1\_MovingAverage1Ch\_1 (FB)
        - FB\_N1\_M2\_MinMaxAvgInterval1Ch\_1 (FB)
        - FB\_N1\_M3\_EdgeCounter1Ch\_1 (FB)
        - FB\_N1\_M4\_ThresholdClassifier1Ch\_1 (FB)
      - FB\_N1\_Network1 (FB)
  - Results:**
    - FB\_Results (FB)
    - I\_Results
  - Analysis:** FB\_Analysis (FB)
  - Main:**
    - MAIN (PRG)
    - MAIN\_Analytics (PRG)
    - A\_Reset
  - Task:**
    - AnalyticsTask (AnalyticsTask)
    - MAIN\_Analytics
  - PlcTask (PlcTask)
  - AnalyticsProject Instance

#### Task:

Für die Analytics-Analyse wird ein separater Task erzeugt.

#### StreamHelper:

Wenn eine oder mehrere Datenquellen vom Typ MQTT-Binärstrom sind, erstellt die Codegenerierung eine Instanz eines StreamHelper-Objekts, um die eingehenden Binärstrommuster zu verarbeiten.

**DataTypes:**

Die Datentypen werden für die Analyse erstellt. Sie enthalten STRUCTs für die Reset-Funktion oder Ergebnisverarbeitung und ENUMs, um die verschiedenen Komponenten auszuwählen.

**HMI GVL:**

Um bequem Ein- und Ausgänge der Module mit dem HMI Dashboard zu mappen, werden ausgewählte Variablen als globale Variable generiert.

**DataSource/M2M Mapping:**

Der FB DataSource verwaltet den Empfang der Eingangswerte der verschiedenen Quellen. In der OUTPUT-Deklaration finden Sie alle konfigurierten Eingänge. Der FB ValueMapping\_M2M verwaltet das Wertemapping zwischen den Modulen (M2M – Module to Module) vom Modul INPUTs zum Modul OUTPUTs.

**Network:**

Alle Module werden in einem spezifischen Netzwerk sortiert, um eine bessere Übersicht und Struktur der konfigurierten Analyse zu erreichen.

**Modules:**

Die Modul-FBs enthalten alle Eingänge und Ausgänge der konfigurierten Module aus dem Workbench-Konfigurator. Es ist auch möglich, die Module während der Laufzeit neu zu konfigurieren. Dazu müssen Sie lediglich den Parameter ändern und dann den Neukonfigurierungsprozess mit einer steigenden Flanke am INPUT bReconfigure starten.

**Results:**

Wenn Ergebnisse der Analyse gespeichert oder gestreamt werden müssen, verwaltet dies der FB Results und streamt die ausgewählten Variablen zum Message-Broker oder speichert die Daten in die Analytics-Binärdatei.

**Analyse:**

Im FB Analysis ist die gesamte Analyseroutine definiert. Alle konfigurierten Netzwerke mit ihren Modulen und Fehlerbehandlung werden erstellt.

**MAIN:**

Im Programm MAIN\_Analytics wird der FB Analysis aufgerufen. Das Programm ist dem separaten Task zugewiesen.

Es ist auch möglich, einzelne Module, ganze Netzwerke oder alle definierten Netzwerke mit nur einem steigendem Flag zurückzusetzen. Zunächst müssen Sie die Komponente auswählen, die zurückgesetzt werden soll. Dann startet eine steigende Flanke am INPUT bReset den Reset-Vorgang.

Alle Reset-Aufrufe sind in der Aktion A\_Reset definiert.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics Version >= 3.1.0.0

**6.1.2.1.1 FB\_DataSource**

Der FB DataSource verwaltet den Empfang der Eingangswerte der verschiedenen Quellen. In der OUTPUT-Deklaration finden Sie alle konfigurierten Eingänge.

**Syntax**

Definition:

```
FUNCTION_BLOCK FB_T[n]_DataSource IMPLEMENTS I_T[n]_DataSource
VAR
END_VAR
```

### Methoden

Name	Definitionsort	Beschreibung
<a href="#">Call</a>  <a href="#">56</a>	Local	Methode für Hintergrundkommunikation mit dem TwinCAT-Treiber. Die Methode muss zyklisch aufgerufen werden.
<a href="#">GetData</a>  <a href="#">56</a>	Local	Methode zum Abrufen der Daten des spezifizierten Elements.

### Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

#### 6.1.2.1.1.1 GetData

##### Syntax

```
METHOD GetData : BOOL
VAR_INPUT
  nElement : UDINT;
END_VAR
```

### Eingänge

Name	Typ	Beschreibung
nElement	UDINT	Element-ID, um das spezifische Beispiel zu erhalten

### Rückgabewert

Name	Typ	Beschreibung
GetData	BOOL	Ist TRUE, wenn ein neues Element ausgewählt ist

#### 6.1.2.1.1.2 Call

##### Syntax

```
METHOD Call : BOOL
```

### Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

#### 6.1.2.1.2 FB\_Network

Alle Module werden in einem spezifischen Netzwerk sortiert, um eine bessere Übersicht und Struktur der konfigurierten Analyse zu erreichen.

## Syntax

Definition:

```
FUNCTION_BLOCK FB_N[n]_[Network1]
VAR_INPUT
    [module FBs]
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
END_VAR
```

### Eingänge

Name	Typ	Beschreibung
Modul-FBs		Funktionsbausteine der konfigurierten Module.

### Ausgänge

Name	Typ	Beschreibung
bError	BOOL	Wird TRUE, sobald eine Fehlersituation auftritt.
ipTcResult	I_TcMessage	Nachrichtenschnittstelle des TwinCAT 3 EventLoggers, die nähere Informationen über den Rückgabewert bietet.

### Methoden

Name	Definitionsart	Beschreibung
<a href="#">Call</a> [ <a href="#">▶ 61</a> ]	Local	Methode für Hintergrundkommunikation. Die Methode muss zyklisch aufgerufen werden.
<a href="#">Reset</a> [ <a href="#">▶ 62</a> ]	Local	Zurücksetzen des Netzwerks mit allen Submodulen.
<a href="#">ValueMapping</a> [ <a href="#">▶ 57</a> ]	Local	Zuordnung der Eingangswerte zu den verschiedenen Moduleingängen.
<a href="#">SetHMIValues</a> [ <a href="#">▶ 58</a> ]	Local	Optional: Mapping der Ein-/Ausgänge der Module zu der globalen HMI-Variable.

## Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

### 6.1.2.1.2.1 ValueMapping

#### Syntax

```
METHOD ValueMapping : BOOL
VAR_INPUT
    ipDataSource : I_T[n]_DataSource;
END_VAR
```

### Eingänge

Name	Typ	Beschreibung
ipDataSource	I_T[n]_DataSource	Daten für die Analyse von der spezifischen Datenquelle

### Rückgabewert

Name	Typ	Beschreibung
ValueMapping	BOOL	

## 6.1.2.1.2.2 SetHMIValues

### Syntax

```
METHOD SetHMIValues : BOOL
VAR_INPUT
    pHMI_N[n]_[Network1] : POINTER TO ST_HMI_N[n]_[Network1];
END_VAR
```

### Eingänge

Name	Typ	Beschreibung
pHMI_N[n]_[Network1]	POINTER TO ST_HMI_N[n]_[Network1]	Zeiger zur globalen HMI-Struktur

### Rückgabewert

Name	Typ	Beschreibung
SetHMIValues	BOOL	Ist TRUE, wenn abgeschlossen

## 6.1.2.1.2.3 Reset

### Syntax

```
METHOD Reset : BOOL
VAR
END_VAR
```

### Rückgabewert

Name	Typ	Beschreibung
Reset	BOOL	Ist TRUE, wenn abgeschlossen

## 6.1.2.1.2.4 Call

### Syntax

```
METHOD Call : BOOL
VAR_INPUT
    ipDataSource: I_T[n]_DataSource;
    [ipValueMapping_M2M: I_ValueMapping_M2M;]
END_VAR
```

### Eingänge

Name	Typ	Beschreibung
ipDataSource	I_T[n]_DataSource	Daten für die Analyse.
ipValueMapping_M2M	I_ValueMapping_M2M	Optional: Notwendig für das Mapping von Werten zwischen Modulen

 **Rückgabewert**

Name	Typ	Beschreibung
Call	BOOL	

**6.1.2.1.3 FB\_Module**

Die Modul-FBs enthalten alle Eingänge und Ausgänge der konfigurierten Module aus dem Workbench-Konfigurator. Es ist auch möglich, die Module während der Laufzeit neu zu konfigurieren. Dazu müssen Sie lediglich den Parameter ändern und dann den Neukonfigurierungsprozess mit einer steigenden Flanke am INPUT bReconfigure starten.

**Syntax**

Definition:

```
FUNCTION_BLOCK FB_N[n]_M[n]_[Module]
VAR_INPUT
    [module inputs]
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
    [module outputs]
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
Module inputs		Eingänge des ausgewählten Moduls.

 **Ausgänge**

Name	Typ	Beschreibung
bError	BOOL	Wird TRUE, sobald eine Fehlersituation auftritt.
ipTcResult	I_TcMessage	Nachrichtenschnittstelle des TwinCAT 3 EventLoggers, die nähere Informationen über den Rückgabewert bietet.
Module outputs		Ausgänge des ausgewählten Moduls.

 **Methoden**

Name	Definitionsart	Beschreibung
<a href="#">Call</a> [ <a href="#">▶ 60</a> ]	Local	Methode für Hintergrundkommunikation. Die Methode muss zyklisch aufgerufen werden.
<a href="#">Reset</a> [ <a href="#">▶ 60</a> ]	Local	Zurücksetzen des Moduls.
<a href="#">SetHMI</a> [ <a href="#">▶ 60</a> ]	Local	Optional: Setzt die Ein-/Ausgänge auf die globalen HMI-Strukturen.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

### 6.1.2.1.3.1 Call

#### Syntax

```
METHOD Call : BOOL
VAR
END_VAR
```

#### Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

### 6.1.2.1.3.2 Reset

#### Syntax

```
METHOD Reset : BOOL
VAR
END_VAR
```

#### Rückgabewert

Name	Typ	Beschreibung
Reset	BOOL	Ist TRUE, wenn abgeschlossen

### 6.1.2.1.3.3 SetHMI

#### Syntax

```
METHOD SetHMIValues : BOOL
VAR_INPUT
    pHMI_N[n]_[Network1] : POINTER TO ST_HMI_N[n]_[Network1];
END_VAR
```

#### Eingänge

Name	Typ	Beschreibung
pHMI_N[n]_[Network1]	POINTER TO ST_HMI_N[n]_[Network1]	Zeiger zur globalen HMI-Struktur

#### Rückgabewert

Name	Typ	Beschreibung
SetHMI	BOOL	Ist TRUE, wenn abgeschlossen

### 6.1.2.1.4 FB\_Analysis

Im FB Analysis ist die gesamte Analyseroutine definiert. Alle konfigurierten Netzwerke mit ihren Modulen und Fehlerbehandlung werden erstellt.

#### Syntax

Definition:

```
FUNCTION_BLOCK FB_Analysis
VAR_INPUT
    [network FBs]
END_VAR
VAR_OUTPUT
    bError: BOOL;
```

```
ipTcResult: I_TcMessage;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
Network FBs		Funktionsbausteine der konfigurierten Netzwerke.

 **Ausgänge**

Name	Typ	Beschreibung
bError	BOOL	Wird TRUE, sobald eine Fehlersituation auftritt.
ipTcResult	I_TcMessage	Nachrichtenschnittstelle des TwinCAT 3 EventLoggers, die nähere Informationen über den Rückgabewert bietet.

 **Methoden**

Name	Definitionsart	Beschreibung
Call <a href="#">[▶ 61]</a>	Local	Methode für Hintergrundkommunikation mit dem TwinCAT-Treiber. Die Methode muss zyklisch aufgerufen werden.
Reset <a href="#">[▶ 62]</a>	Local	Zurücksetzen der gesamten Analyse.
ResultStream <a href="#">[▶ 62]</a>	Local	Optional: Wenn ein Ergebnis-Stream erstellt werden muss.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

**6.1.2.1.4.1 Call**

**Syntax**

```
METHOD Call : BOOL
VAR_INPUT
    ipDataSource: I_T[n]_DataSource;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
ipDataSource	I_T[n]_DataSource	Daten für die Analyse.

 **Rückgabewert**

Name	Typ	Beschreibung
Call	BOOL	

### 6.1.2.1.4.2 Reset

#### Syntax

```
METHOD Reset : BOOL
VAR_IN_OUT
    stReset: ST_AnalysisReset;
END_VAR
```

#### Eingänge

Name	Typ	Beschreibung
stReset	ST_AnalysisReset	Struktur, um zu definieren, welches Modul oder Netzwerk zurückgesetzt werden soll.

#### Rückgabewert

Name	Typ	Beschreibung
Reset	BOOL	Ist TRUE, wenn abgeschlossen

### 6.1.2.1.4.3 ResultStream

#### Syntax

```
METHOD ResultStream : BOOL
VAR_INPUT
    ipResults: I_Results;
END_VAR
```

#### Eingänge

Name	Typ	Beschreibung
ipResults	I_Results	Schnittstellenzeiger auf den FB Results

#### Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

### 6.1.2.1.5 FB\_Results

Wenn Ergebnisse der Analyse gespeichert oder gestreamt werden müssen, verwaltet dies der FB Results und streamt die ausgewählten Variablen zum Message-Broker oder speichert die Daten in die Analytics-Binärdatei.

#### Syntax

Definition:

```
FUNCTION_BLOCK FB_Results
VAR_OUTPUT
    stResults: ST_Results;
END_VAR
```

 **Ausgänge**

Name	Typ	Beschreibung
stResults	ST_Results	Ergebnisstruktur, die alle Elemente des Ergebnis-Streams enthält.

 **Methoden**

Name	Definitionsort	Beschreibung
Call [ <a href="#">▶ 63</a> ]	Local	Methode für Hintergrundkommunikation. Die Methode muss zyklisch aufgerufen werden.
AddResult [ <a href="#">▶ 63</a> ]	Local	Fügt dem Ergebnis-Stream ein Muster hinzu.
SendResults [ <a href="#">▶ 64</a> ]	Local	Sendet alle gepufferten Muster des Ergebnis-Streams.
Release [ <a href="#">▶ 64</a> ]	Local	Schließt den Stream oder die Datei des Ergebnis-Streams.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

**6.1.2.1.5.1 Call**

**Syntax**

```
METHOD Call : BOOL
VAR
END_VAR
```

 **Rückgabewert**

Name	Typ	Beschreibung
Call	BOOL	

**6.1.2.1.5.2 AddResult**

**Syntax**

```
METHOD AddResult : BOOL
VAR_INPUT
    tTimestamp: ULINT;
    stSample: ST_Results;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
tTimestamp	ULINT	Zeitstempel des Samples
stSample	ST_Results	Sample-Struktur

 **Rückgabewert**

Name	Typ	Beschreibung
AddResult	BOOL	

### 6.1.2.1.5.3 SendResults

#### Syntax

```
METHOD SendResults : BOOL
VAR
END_VAR
```

#### Rückgabewert

Name	Typ	Beschreibung
SendResults	BOOL	

### 6.1.2.1.5.4 Release

#### Syntax

```
METHOD Release : BOOL
VAR
END_VAR
```

#### Rückgabewert

Name	Typ	Beschreibung
Release	BOOL	

### 6.1.2.1.6 MAIN\_Analytics

Im Programm MAIN\_Analytics wird der FB Analysis aufgerufen. Das Programm ist dem separaten Task zugewiesen.

Es ist auch möglich, einzelne Module, ganze Netzwerke oder alle definierten Netzwerke mit nur einem steigendem Flag zurückzusetzen. Zunächst müssen Sie die Komponente auswählen, die zurückgesetzt werden soll. Dann startet eine steigende Flanke am INPUT bReset den Reset-Vorgang.

Alle Reset-Aufrufe sind in der Aktion A\_Reset definiert.

#### Syntax

Definition:

```
PROGRAM MAIN_Analytics
VAR_INPUT
    stReset: ST_AnalysisReset;
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
END_VAR
```

#### Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

### 6.1.2.2 Codeversion 3.0

**Tasks:**

Für die Analytics-Analyse und für jede Konfiguration einer Virtual Input Source wird ein separater Task erzeugt.

**StreamHelper:**

Für jede Datenquellen vom Typ MQTT-Binärstrom, erstellt die Codegenerierung eine Instanz eines Stream Helper-Objekts, um die eingehenden Binärstrommuster zu verarbeiten.

**DataTypes:**

Die Datentypen werden für die Analyse erstellt. Sie enthalten STRUCTs für die Reset-Funktion oder Ergebnisverarbeitung und ENUMs, um die verschiedenen Komponenten auszuwählen.

**GVLs:**

Um bequem Ein- und Ausgänge der Module mit dem HMI Dashboard zu mappen, werden ausgewählte Variablen als globale Variable generiert. Zusätzlich werden die Data Source Baustein Instanzen und verschiedene Parameter als Globale Variable generiert.

**VirtualInputSource / DataSource / M2M Mapping:**

Die Virtual Input Source Interfaces abstrahieren die Data Source Symbole von der Analyse. Der FB DataSource verwaltet den Empfang der Eingangswerte der verschiedenen Quellen. In der OUTPUT-Deklaration finden Sie alle konfigurierten Eingänge. Der FB ValueMapping\_M2M verwaltet das Wertemapping zwischen den Modulen (M2M – Module to Module) vom Modul INPUTs zum Modul OUTPUTs.

**Network:**

Alle Module werden in einem spezifischen Netzwerk sortiert, um eine bessere Übersicht und Struktur der konfigurierten Analyse zu erreichen.

**Modules:**

Die Modul-FBs enthalten alle Eingänge und Ausgänge der konfigurierten Module aus dem Workbench-Konfigurator. Es ist auch möglich, die Module während der Laufzeit neu zu konfigurieren. Dazu müssen Sie lediglich den Parameter ändern und dann den Neukonfigurierungsprozess mit einer steigenden Flanke am INPUT bReconfigure starten.

**Results:**

Wenn Ergebnisse der Analyse gespeichert oder gestreamt werden müssen, verwaltet dies der FB Results und streamt die ausgewählten Variablen zum Message-Broker im Binär oder Json Format, oder speichert die Daten lokal in eine Analytics-Binärdatei.

**Analysis:**

Im FB Analysis ist die gesamte Analyseroutine definiert. Alle konfigurierten Netzwerke mit ihren Modulen und Fehlerbehandlung werden erstellt.

**MAIN PRGs:**

In dem Programm MAIN\_Analytics werden die DataSource FBs aufgerufen, die Reset Funktion verwaltet und ggf. die Werte mit dem HMI Dashboard gemapped. Das Programm ist einem separaten Task zugewiesen.

In den Programmen MAIN\_Analytics\_Vx\_Cx wird der FB Analysis aufgerufen. Die Programme sind jeweils einem separaten Task zugewiesen.

Es ist auch möglich, einzelne Module, ganze Netzwerke oder alle definierten Netzwerke mit nur einem steigendem Flag zurückzusetzen. Zunächst müssen Sie die Komponente auswählen, die zurückgesetzt werden soll. Dann startet eine steigende Flanke am INPUT bReset den Reset-Vorgang.

Alle Reset-Aufrufe sind in der Aktion A\_Reset definiert.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics Version >= 3.1.0.0

### 6.1.2.2.1 FB\_DataSource

Der FB DataSource verwaltet den Empfang der Eingangswerte der verschiedenen Quellen. In der OUTPUT-Deklaration finden Sie alle konfigurierten Eingänge.

#### Syntax

Definition:

```
FUNCTION_BLOCK FB_T[n]_DataSource IMPLEMENTS I_DataSource, I_V[n]_Virtual_Input_Source
VAR_OUTPUT
eDataState: E_DataSourceState;
END_VAR
```

#### Methoden

Name	Definitionsort	Beschreibung
<a href="#">Call</a> [ <a href="#">▶ 67</a> ]	Local	Methode für Hintergrundkommunikation mit dem TwinCAT-Treiber. Die Methode muss zyklisch aufgerufen werden.
<a href="#">GetData</a> [ <a href="#">▶ 67</a> ]	Local	Methode zum Abrufen der Daten des spezifizierten Elements.
<a href="#">GetDataOversampling</a> [ <a href="#">▶ 68</a> ]	Local	Methode zum Abrufen der Oversampling Daten des spezifizierten Elements.
<a href="#">NewDataAvailable</a> [ <a href="#">▶ 68</a> ]	Local	Methode zum Prüfen ob neue Daten verfügbar sind.
<a href="#">HistoricalCtrl</a> [ <a href="#">▶ 68</a> ]	Local	Methode zum Abrufen von historischen Daten.
<a href="#">UpdateRecordList</a> [ <a href="#">▶ 69</a> ]	Local	Methode zum Aktualisieren der Historischen Aufnahmeliste.

#### Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

### 6.1.2.2.1.1 GetData

#### Syntax

```
METHOD GetData : BOOL
VAR_INPUT
nElement : UDINT;
END_VAR
```

#### Eingänge

Name	Typ	Beschreibung
nElement	UDINT	Element-ID, um das spezifische Beispiel zu erhalten.

#### Rückgabewert

Name	Typ	Beschreibung
GetData	BOOL	Ist TRUE, wenn ein neues Element ausgewählt ist.

### 6.1.2.2.1.2 Call

#### Syntax

```
METHOD Call : BOOL
```

### Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

## 6.1.2.2.1.3 GetDataOversampling

### Syntax

```
METHOD GetDataOversampling : BOOL
VAR_INPUT
nElement : UDINT;
nSample : UDINT;
END_VAR
```

### Eingänge

Name	Typ	Beschreibung
nElement	UDINT	Element-ID, um das spezifische Sample-Element zu erhalten.
nSample	UDINT	Sample-ID, um das spezifische Sample zu erhalten.

### Rückgabewert

Name	Typ	Beschreibung
GetDataOversampling	BOOL	Ist TRUE, wenn ein neues Element ausgewählt ist.

## 6.1.2.2.1.4 NewDataAvailable

### Syntax

```
METHOD NewDataAvailable : BOOL
VAR_INPUT
nLastDataHandle : ULINT;
END_VAR
```

### Eingänge

Name	Typ	Beschreibung
nLastDataHandle	ULINT	Handle vom zuletzt abgeholten Datenpaket.

### Rückgabewert

Name	Typ	Beschreibung
NewDataAvailable	BOOL	Ist TRUE, wenn neue Daten verfügbar sind.

## 6.1.2.2.1.5 HistoricalCtrl

### Syntax

```
METHOD HistoricalCtrl : BOOL
VAR_INPUT
stCtrl : REFERENCE TO ST_HMI_DataSourceCtrl;
stHistStreamInfo : REFERENCE TO ST_HMI_DataSourceHist;
stRecordInfo : REFERENCE TO ST_HMI_DataSourceHistRecordInfo;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
stCtrl	ST_HMI_DataSourceCtrl	
stHistStreamInfo	ST_HMI_DataSourceHist	
stRecordInfo	ST_HMI_DataSourceHistRecordInfo	

 **Rückgabewert**

Name	Typ	Beschreibung
HistoricalCtrl	BOOL	

### 6.1.2.2.1.6 UpdateRecordList

**Syntax**

```
METHOD UpdateRecordList : BOOL
VAR_INPUT
    stCtrl : REFERENCE TO ST_HMI_DataSourceCtrl;
    stHistStreamInfo : REFERENCE TO ST_HMI_DataSourceHist;
    sStreamSystemID : GUID;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
stCtrl	ST_HMI_DataSourceCtrl	
stHistStreamInfo	ST_HMI_DataSourceHist	
sStreamSystemID	GUID	

 **Rückgabewert**

Name	Typ	Beschreibung
UpdateRecordList	BOOL	

### 6.1.2.2.2 FB\_Network

Alle Module werden in einem spezifischen Netzwerk sortiert, um eine bessere Übersicht und Struktur der konfigurierten Analyse zu erreichen.

**Syntax**

Definition:

```
FUNCTION_BLOCK FB_N[n]_[Network1]
VAR_INPUT
    [module FBs]
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
Modul-FBs		Funktionsbausteine der konfigurierten Module.

### Ausgänge

Name	Typ	Beschreibung
bError	BOOL	Wird TRUE, sobald eine Fehlersituation auftritt.
ipTcResult	I_TcMessage	Nachrichtenschnittstelle des TwinCAT 3 EventLoggers, die weitere Informationen über den Rückgabewert bietet.

### Methoden

Name	Definitionsort	Beschreibung
<a href="#">Call</a> [ <a href="#">▶ 74</a> ]	Local	Methode für Hintergrundkommunikation. Die Methode muss zyklisch aufgerufen werden.
<a href="#">Reset</a> [ <a href="#">▶ 74</a> ]	Local	Zurücksetzen des Netzwerks mit allen Submodulen.
<a href="#">ValueMapping</a> [ <a href="#">▶ 70</a> ]	Local	Zuordnung der Eingangswerte zu den verschiedenen Moduleingängen.
<a href="#">SetHMIValues</a> [ <a href="#">▶ 70</a> ]	Local	Optional: Mapping der Ein-/Ausgänge der Module zu der globalen HMI-Variable.

### Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

## 6.1.2.2.2.1 ValueMapping

### Syntax

```
METHOD ValueMapping : BOOL
VAR_INPUT
    ipVirtual_Input_Source : I_V[n]_Virtual_Input_Source;
END_VAR
```

### Eingänge

Name	Typ	Beschreibung
ipVirtual_Input_Source	I_V[n]_Virtual_Input_Source	Daten für die Analyse von der spezifischen Datenquelle.

### Rückgabewert

Name	Typ	Beschreibung
ValueMapping	BOOL	

## 6.1.2.2.2.2 SetHMIValues

### Syntax

```
METHOD SetHMIValues : BOOL
VAR_INPUT
    pHMI_N[n]_[Network1] : POINTER TO ST_HMI_N[n]_[Network1];
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
pHMI_N[n]_[Network1]	POINTER TO ST_HMI_N[n]_[Network1]	Zeiger zur globalen HMI-Struktur.

 **Rückgabewert**

Name	Typ	Beschreibung
SetHMIValues	BOOL	Ist TRUE, wenn abgeschlossen.

**6.1.2.2.2.3 Reset**

**Syntax**

```
METHOD Reset : BOOL
VAR
END_VAR
```

 **Rückgabewert**

Name	Typ	Beschreibung
Reset	BOOL	Ist TRUE, wenn abgeschlossen.

**6.1.2.2.2.4 Call**

**Syntax**

```
METHOD Call : BOOL
VAR_INPUT
    ipVirtual_Input_Source: I_V[n]_Virtual_Input_Source;
    [ipValueMapping_M2M: I_ValueMapping_M2M;]
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
ipVirtual_Input_Source	I_T[n]_Virtual_Input_Source	Daten für die Analyse.
ipValueMapping_M2M	I_ValueMapping_M2M	Optional: Notwendig für das Mapping von Werten zwischen Modulen.

 **Rückgabewert**

Name	Typ	Beschreibung
Call	BOOL	

**6.1.2.2.3 FB\_Module**

Die Modul-FBs enthalten alle Eingänge und Ausgänge der konfigurierten Module aus dem Workbench-Konfigurator. Es ist auch möglich, die Module während der Laufzeit neu zu konfigurieren. Dazu müssen Sie lediglich den Parameter ändern und dann den Neukonfigurierungsprozess mit einer steigenden Flanke am INPUT bReconfigure starten.

**Syntax**

Definition:

```

FUNCTION_BLOCK FB_N[n]_M[n]_[Module]
VAR_INPUT
    [module inputs]
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
    [module outputs]
END_VAR

```

### Eingänge

Name	Typ	Beschreibung
Module inputs		Eingänge des ausgewählten Moduls.

### Ausgänge

Name	Typ	Beschreibung
bError	BOOL	Wird TRUE, sobald eine Fehlersituation auftritt.
ipTcResult	I_TcMessage	Nachrichtenschnittstelle des TwinCAT 3 EventLoggers, die weitere Informationen über den Rückgabewert bietet.
Module outputs		Ausgänge des ausgewählten Moduls.

### Methoden

Name	Definitionsart	Beschreibung
Call <a href="#">[▶ 72]</a>	Local	Methode für Hintergrundkommunikation. Die Methode muss zyklisch aufgerufen werden.
Reset <a href="#">[▶ 73]</a>	Local	Zurücksetzen des Moduls.
SetHMI <a href="#">[▶ 73]</a>	Local	Optional: Setzt die Ein-/Ausgänge auf die globalen HMI-Strukturen.

### Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

## 6.1.2.2.3.1 Call

### Syntax

```

METHOD Call : BOOL
VAR_INPUT
    ipVirtual_Input_Source : I_V[n]_Virtual_Input_Source;
    [ipValueMapping_M2M : I_ValueMapping_M2M;]
END_VAR

```

### Eingänge

Name	Typ	Beschreibung
ipVirtual_Input_Source	I_V[n]_Virtual_Input_Source	Daten für die Analyse.
ipValueMapping_M2M	I_ValueMapping_M2M	Optional: Notwendig für das Mapping von Werten zwischen Modulen.

 Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

### 6.1.2.2.3.2 Reset

**Syntax**

```
METHOD Reset : BOOL
VAR
END_VAR
```

 Rückgabewert

Name	Typ	Beschreibung
Reset	BOOL	Ist TRUE, wenn abgeschlossen.

### 6.1.2.2.3.3 SetHMI

**Syntax**

```
METHOD SetHMIValues : BOOL
VAR_INPUT
    pHMI_N[n]_[Network1] : POINTER TO ST_HMI_N[n]_[Network1];
END_VAR
```

 Eingänge

Name	Typ	Beschreibung
pHMI_N[n]_[Network1]	POINTER TO ST_HMI_N[n]_[Network1]	Zeiger zur globalen HMI-Struktur.

 Rückgabewert

Name	Typ	Beschreibung
SetHMI	BOOL	Ist TRUE, wenn abgeschlossen.

### 6.1.2.2.4 FB\_Analysis

Im FB Analysis ist die gesamte Analyseroutine definiert. Alle konfigurierten Netzwerke mit ihren Modulen und Fehlerbehandlung werden erstellt.

**Syntax**

Definition:

```
FUNCTION_BLOCK FB_Analysis
VAR_INPUT
    [network FBs]
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
END_VAR
```

### Eingänge

Name	Typ	Beschreibung
Network FBs		Funktionsbausteine der konfigurierten Netzwerke.

### Ausgänge

Name	Typ	Beschreibung
bError	BOOL	Wird TRUE, sobald eine Fehlersituation auftritt.
ipTcResult	I_TcMessage	Nachrichtenschnittstelle des TwinCAT 3 EventLoggers, die weitere Informationen über den Rückgabewert bietet.

### Methoden

Name	Definitionsart	Beschreibung
Call [ <a href="#">▶ 74</a> ]	Local	Methode für Hintergrundkommunikation mit dem TwinCAT-Treiber. Die Methode muss zyklisch aufgerufen werden.
Reset [ <a href="#">▶ 74</a> ]	Local	Zurücksetzen der gesamten Analyse.
ResultStream [ <a href="#">▶ 75</a> ]	Local	Optional: Wenn ein Ergebnis-Stream erstellt werden muss.

### Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

## 6.1.2.2.4.1 Call

### Syntax

```
METHOD Call : BOOL
VAR_INPUT
    ipVirtual_Input_Source: I_V[n]_Virtual_Input_Source;
END_VAR
```

### Eingänge

Name	Typ	Beschreibung
ipVirtual_Input_Source	I_V[n]_Virtual_Input_Source	Daten für die Analyse.

### Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

## 6.1.2.2.4.2 Reset

### Syntax

```
METHOD Reset : BOOL
VAR_IN_OUT
    stReset: ST_AnalysisReset;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
stReset	ST_AnalysisReset	Struktur, um zu definieren, welches Modul oder Netzwerk zurückgesetzt werden soll.

 **Rückgabewert**

Name	Typ	Beschreibung
Reset	BOOL	Ist TRUE, wenn abgeschlossen.

**6.1.2.2.4.3 ResultStream**

**Syntax**

```
METHOD ResultStream : BOOL
VAR_INPUT
    ipResults: I_Results;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
ipResults	I_Results	Schnittstellenzeiger auf den FB Results.

 **Rückgabewert**

Name	Typ	Beschreibung
Call	BOOL	

**6.1.2.2.5 FB\_Results**

Wenn Ergebnisse der Analyse gespeichert oder gestreamt werden müssen, verwaltet dies der FB Results und streamt die ausgewählten Variablen zum Message-Broker oder speichert die Daten in die Analytics-Binärdatei.

**Syntax**

Definition:

```
FUNCTION_BLOCK FB_Results
VAR_OUTPUT
    stResults: ST_Results;
END_VAR
```

 **Ausgänge**

Name	Typ	Beschreibung
stResults	ST_Results	Ergebnisstruktur, die alle Elemente des Ergebnis-Streams enthält.

## Methoden

Name	Definitionsart	Beschreibung
<a href="#">Call [▶ 76]</a>	Local	Methode für Hintergrundkommunikation. Die Methode muss zyklisch aufgerufen werden.
<a href="#">AddResult [▶ 76]</a>	Local	Fügt dem Ergebnis-Stream ein Muster hinzu.
<a href="#">SendResults [▶ 76]</a>	Local	Sendet alle gepufferten Muster des Ergebnis-Streams.
<a href="#">Release [▶ 77]</a>	Local	Schließt den Stream oder die Datei des Ergebnis-Streams.

## Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

### 6.1.2.2.5.1 Call

#### Syntax

```
METHOD Call : BOOL
VAR
END_VAR
```

#### Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

### 6.1.2.2.5.2 AddResult

#### Syntax

```
METHOD AddResult : BOOL
VAR_INPUT
    tTimestamp: ULINT;
    stSample: ST_Results;
END_VAR
```

#### Eingänge

Name	Typ	Beschreibung
tTimestamp	ULINT	Zeitstempel des Samples
stSample	ST_Results	Sample-Struktur

#### Rückgabewert

Name	Typ	Beschreibung
AddResult	BOOL	

### 6.1.2.2.5.3 SendResults

#### Syntax

```
METHOD SendResults : BOOL
VAR
END_VAR
```

 Rückgabewert

Name	Typ	Beschreibung
SendResults	BOOL	

### 6.1.2.2.5.4 Release

**Syntax**

```
METHOD Release : BOOL
VAR
END_VAR
```

 Rückgabewert

Name	Typ	Beschreibung
Release	BOOL	

### 6.1.2.2.6 MAIN\_Analytics

Im Programm MAIN\_Analytics werden die DataSource FBs aufgerufen, die Reset Funktion gemanaged und ggf. die Werte mit dem HMI Dashboard gemapped. Das Programm ist einem separaten Task zugewiesen.

**Syntax**

Definition:

```
PROGRAM MAIN_Analytics
VAR_INPUT
    sCurrentStreamSystemID: GUID;
    stHistStreamInfo: REFERENCE TO ST_HMI_DataSourceHist;
    stHistRecordInfo: REFERENCE TO ST_HMI_DataSourceHistRecordInfo;
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
END_VAR
```

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

### 6.1.2.2.7 MAIN\_Analytics\_V[n]\_C[n]

Im Programm MAIN\_Analytics\_V[n]\_C[n] wird der FB Analysis aufgerufen. Das Programm ist dem separaten Task zugewiesen.

Es ist auch möglich, einzelne Module, ganze Netzwerke oder alle definierten Netzwerke mit nur einem steigendem Flag zurückzusetzen. Zunächst müssen Sie die Komponente auswählen, die zurückgesetzt werden soll. Dann startet eine steigende Flanke am INPUT bReset den Reset-Vorgang.

Alle Reset-Aufrufe sind in der Aktion A\_Reset definiert.

Das Mapping der HMI Werte wird in der Aktion A\_MapToHMI vorgenommen.

**Syntax**

Definition:

```
PROGRAM MAIN_Analytics_V[n]_C[n]
VAR_INPUT
    stReset: ST_AnalysisReset;
```

```
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
    nLastDataHandle: ULINT;
END_VAR
```

### Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

### 6.1.2.3 Codeversion 4.x

**Tasks:**

Für die Analytics-Analyse und für jede Konfiguration einer Virtual Input Source wird ein separater Task erzeugt.

**StreamHelper:**

Für jede Datenquellen vom Typ MQTT-Binärstrom, erstellt die Codegenerierung eine Instanz eines Stream Helper-Objekts, um die eingehenden Binärstrommuster zu verarbeiten.

**DataTypes:**

Die Datentypen werden für die Analyse erstellt. Sie enthalten STRUCTs für die Reset-Funktion oder Ergebnisverarbeitung und ENUMs, um die verschiedenen Komponenten auszuwählen.

**GVLs:**

Um bequem Ein- und Ausgänge der Module mit dem HMI Dashboard zu mappen, werden ausgewählte Variablen als globale Variable generiert. Zusätzlich werden die Data Source Baustein Instanzen und verschiedene Parameter als Globale Variable generiert.

**ClosedNetwork:**

Die ClosedNetwork FBs werden einmal mit allen Subnetzwerken und Modulen generiert. Sie können in der Analyse mehrmalig instanziiert werden. Auf diese Weise kann der generierte Code reduziert und vereinfacht werden.

**VirtuallInputSource / DataSource:**

Die VirtuallInputSource Interfaces abstrahieren die DataSource Symbole von der Analyse. Der FB DataSource verwaltet den Empfang der Eingangswerte der verschiedenen Quellen. In der OUTPUT-Deklaration finden Sie alle konfigurierten Eingänge.

**Network:**

Alle Module werden in einem spezifischen Netzwerk sortiert, um eine bessere Übersicht und Struktur der konfigurierten Analyse zu erreichen.

**Modules:**

Die Modul-FBs enthalten alle Eingänge und Ausgänge der konfigurierten Module aus dem Workbench-Konfigurator. Es ist auch möglich, die Module während der Laufzeit neu zu konfigurieren. Dazu müssen Sie lediglich den Parameter ändern und dann den Neukonfigurierungsprozess mit einer steigenden Flanke am INPUT bReconfigure starten.

**ClosedNetwork Instance:**

In diesem FB wird das entsprechende ClosedNetwork für die Analyse instanziiert. Intern verwendete Module werden in diesem Fall nicht weiter als Modul-FBs generiert.

**Results:**

Wenn Ergebnisse der Analyse gespeichert oder gestreamt werden müssen, verwaltet dies der FB Results und streamt die ausgewählten Variablen zum Message-Broker im Binär oder Json Format, oder speichert die Daten lokal in eine Analytics-Binärdatei.

**Analysis:**

Im FB Analysis ist die gesamte Analyseroutine definiert. Alle konfigurierten Netzwerke mit ihren Modulen und Fehlerbehandlung werden erstellt.

**MAIN PRGs:**

In dem Programm MAIN\_Analytics werden die DataSource FBs aufgerufen, die Reset Funktion verwaltet und ggf. die Werte mit dem HMI Dashboard gemapped. Das Programm ist einem separaten Task zugewiesen.

In den Programmen MAIN\_Analytics\_Vx\_Cx wird der FB Analysis aufgerufen. Die Programme sind jeweils einem separaten Task zugewiesen.

Es ist auch möglich, einzelne Module, ganze Netzwerke oder alle definierten Netzwerke mit nur einem steigendem Flag zurückzusetzen. Zunächst müssen Sie die Komponente auswählen, die zurückgesetzt werden soll. Dann startet eine steigende Flanke am INPUT bReset den Reset-Vorgang.

Alle Reset-Aufrufe sind in der Aktion A\_Reset definiert.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics Version >= 3.1.0.0

**6.1.2.3.1 FB\_DataSource**

Der FB DataSource verwaltet den Empfang der Eingangswerte der verschiedenen Quellen. In der OUTPUT-Deklaration finden Sie alle konfigurierten Eingänge.

**Syntax**

Definition:

```
FUNCTION_BLOCK FB_T[n]_DataSource IMPLEMENTS I_DataSource, I_V[n]_Virtual_Input_Source
VAR_OUTPUT
eDataState: E_DataSourceState;
END_VAR
```

 **Methoden**

Name	Definitionsort	Beschreibung
<a href="#">Call [▶ 82]</a>	Local	Methode für Hintergrundkommunikation mit dem TwinCAT-Treiber. Die Methode muss zyklisch aufgerufen werden.
<a href="#">GetData [▶ 81]</a>	Local	Methode zum Abrufen der Daten des spezifizierten Elements.
<a href="#">GetDataOversampling [▶ 82]</a>	Local	Methode zum Abrufen der Oversampling Daten des spezifizierten Elements
<a href="#">NewDataAvailable [▶ 82]</a>	Local	Methode zum Prüfen ob neue Daten verfügbar sind.
<a href="#">HistoricalCtrl [▶ 83]</a>	Local	Methode zum Abrufen von historischen Daten
<a href="#">UpdateRecordList [▶ 83]</a>	Local	Methode zum Aktualisieren der Historischen Aufnahmeliste.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

**6.1.2.3.1.1 GetData**

**Syntax**

```
METHOD GetData : BOOL
VAR_INPUT
nElement : UDINT;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
nElement	UDINT	Element-ID, um das spezifische Beispiel zu erhalten

### Rückgabewert

Name	Typ	Beschreibung
GetData	BOOL	Ist TRUE, wenn ein neues Element ausgewählt ist

### 6.1.2.3.1.2 Call

#### Syntax

```
METHOD Call : BOOL
```

### Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

### 6.1.2.3.1.3 GetDataOversampling

#### Syntax

```
METHOD GetDataOversampling : BOOL
VAR_INPUT
nElement : UDINT;
nSample : UDINT;
END_VAR
```

### Eingänge

Name	Typ	Beschreibung
nElement	UDINT	Element-ID, um das spezifische Sampleelement zu erhalten
nSample	UDINT	Sample-ID, um das spezifische Sample zu erhalten

### Rückgabewert

Name	Typ	Beschreibung
GetDataOversampling	BOOL	Ist TRUE, wenn ein neues Element ausgewählt ist

### 6.1.2.3.1.4 NewDataAvailable

#### Syntax

```
METHOD NewDataAvailable : BOOL
VAR_INPUT
nLastDataHandle : ULINT;
END_VAR
```

### Eingänge

Name	Typ	Beschreibung
nLastDataHandle	ULINT	Handle vom zuletzt abgeholten Datenpaket

### Rückgabewert

Name	Typ	Beschreibung
NewDataAvailable	BOOL	Ist TRUE, wenn neue Daten verfügbar sind

### 6.1.2.3.1.5 HistoricalCtrl

#### Syntax

```
METHOD HistoricalCtrl : BOOL
VAR_INPUT
    stCtrl : REFERENCE TO ST_HMI_DataSourceCtrl;
    stHistStreamInfo : REFERENCE TO ST_HMI_DataSourceHist;
    stRecordInfo : REFERENCE TO ST_HMI_DataSourceHistRecordInfo;
END_VAR
```

#### Eingänge

Name	Typ	Beschreibung
stCtrl	ST_HMI_DataSourceCtrl	
stHistStreamInfo	ST_HMI_DataSourceHist	
stRecordInfo	ST_HMI_DataSourceHistRecordInfo	

#### Rückgabewert

Name	Typ	Beschreibung
HistoricalCtrl	BOOL	

### 6.1.2.3.1.6 UpdateRecordList

#### Syntax

```
METHOD UpdateRecordList : BOOL
VAR_INPUT
    stCtrl : REFERENCE TO ST_HMI_DataSourceCtrl;
    stHistStreamInfo : REFERENCE TO ST_HMI_DataSourceHist;
    sStreamSystemID : GUID;
END_VAR
```

#### Eingänge

Name	Typ	Beschreibung
stCtrl	ST_HMI_DataSourceCtrl	
stHistStreamInfo	ST_HMI_DataSourceHist	
sStreamSystemID	GUID	

#### Rückgabewert

Name	Typ	Beschreibung
UpdateRecordList	BOOL	

### 6.1.2.3.1.7 NextData

#### Syntax

```
METHOD NextData : BOOL
VAR_INPUT
END_VAR
```

### 6.1.2.3.1.8 NextDataOversample

#### Syntax

```
METHOD GetDataOversampling : BOOL
VAR_INPUT
nMaxOversampling : UDINT;
END_VAR
```

#### Eingänge

Name	Typ	Beschreibung
nMaxOversampling	UDINT	Gibt den maximalen Oversamplingfaktor an.

#### Rückgabewert

Name	Typ	Beschreibung
NextDataOversample	BOOL	Ist TRUE, wenn ein neues Element ausgewählt ist

### 6.1.2.3.2 FB\_Network

Alle Module werden in einem spezifischen Netzwerk sortiert, um eine bessere Übersicht und Struktur der konfigurierten Analyse zu erreichen.

#### Syntax

Definition:

```
FUNCTION_BLOCK FB_N[n]_[Network1]
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
END_VAR
VAR
    [module FBs]
END_VAR
```

#### Ausgänge

Name	Typ	Beschreibung
bError	BOOL	Wird TRUE, sobald eine Fehlersituation auftritt.
ipTcResult	I_TcMessage	Nachrichtenschnittstelle des TwinCAT 3 EventLoggers, die nähere Informationen über den Rückgabewert bietet.

#### Methoden

Name	Definitionsart	Beschreibung
<a href="#">Call [▶ 74]</a>	Local	Methode für Hintergrundkommunikation. Die Methode muss zyklisch aufgerufen werden.
<a href="#">Reset [▶ 74]</a>	Local	Zurücksetzen des Netzwerks mit allen Submodulen.
<a href="#">ValueMapping [▶ 85]</a>	Local	Zuordnung der Eingangswerte zu den verschiedenen Moduleingängen.
<a href="#">SetHMIValues [▶ 85]</a>	Local	Optional: Mapping der Ein-/Ausgänge der Module zu der globalen HMI-Variable.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

**6.1.2.3.2.1 ValueMapping**

**Syntax**

```
METHOD ValueMapping : BOOL
VAR_INPUT
    pAnalysis : POINTER TO FB_Analysis;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
pAnalysis	FB_Analysis	Instanz des Analyse FBs

 **Rückgabewert**

Name	Typ	Beschreibung
ValueMapping	BOOL	

**6.1.2.3.2.2 SetHMIValues**

**Syntax**

```
METHOD SetHMIValues : BOOL
VAR_INPUT
    pHMI_N[n]_[Network1] : POINTER TO ST_HMI_N[n]_[Network1];
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
pHMI_N[n]_[Network1]	POINTER TO ST_HMI_N[n]_[Network1]	Zeiger zur globalen HMI-Struktur

 **Rückgabewert**

Name	Typ	Beschreibung
SetHMIValues	BOOL	Ist TRUE, wenn abgeschlossen

**6.1.2.3.2.3 Call**

**Syntax**

```
METHOD Call : BOOL
VAR_INPUT
    pAnalysis: POINTER TO FB_Analysis;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
pAnalysis	FB_Analysis	Instanz des Analyse FBs.

### Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

#### 6.1.2.3.2.4 Reset

##### Syntax

```
METHOD Reset : BOOL
VAR
END_VAR
```

### Rückgabewert

Name	Typ	Beschreibung
Reset	BOOL	Ist TRUE, wenn abgeschlossen.

#### 6.1.2.3.3 FB\_Module

Die Modul-FBs enthalten alle Eingänge und Ausgänge der konfigurierten Module aus dem Workbench-Konfigurator. Es ist auch möglich, die Module während der Laufzeit neu zu konfigurieren. Dazu müssen Sie lediglich den Parameter ändern und dann den Neukonfigurierungsprozess mit einer steigenden Flanke am INPUT bReconfigure starten.

##### Syntax

Definition:

```
FUNCTION_BLOCK FB_N[n]_M[n]_[Module]
VAR_INPUT
  [module inputs]
END_VAR
VAR_OUTPUT
  bError: BOOL;
  ipTcResult: I_TcMessage;
  [module outputs]
END_VAR
```

### Eingänge

Name	Typ	Beschreibung
Module inputs		Eingänge des ausgewählten Moduls.

### Ausgänge

Name	Typ	Beschreibung
bError	BOOL	Wird TRUE, sobald eine Fehlersituation auftritt.
ipTcResult	I_TcMessage	Nachrichtenschnittstelle des TwinCAT 3 EventLoggers, die nähere Informationen über den Rückgabewert bietet.
Module outputs		Ausgänge des ausgewählten Moduls.

 **Methoden**

Name	Definitionsort	Beschreibung
<a href="#">Call [▶ 87]</a>	Local	Methode für Hintergrundkommunikation. Die Methode muss zyklisch aufgerufen werden.
<a href="#">Reset [▶ 87]</a>	Local	Zurücksetzen des Moduls.
<a href="#">SetHMI [▶ 88]</a>	Local	Optional: Setzt die Ein-/Ausgänge auf die globalen HMI-Strukturen.
<a href="#">GetHMI [▶ 88]</a>	Local	Optional: Setzt die Eingänge der globalen HMI-Strukturen auf die Eingänge des Moduls

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

**6.1.2.3.3.1 Call**

**Syntax**

```
METHOD Call : BOOL
VAR_INPUT
    ipVirtual_Input_Source : I_V[n]_Virtual_Input_Source;
    [ipValueMapping_M2M : I_ValueMapping_M2M;]
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
ipVirtual_Input_Source	I_V[n]_Virtual_Input_Source	Daten für die Analyse
ipValueMapping_M2M	I_ValueMapping_M2M	Optional: Notwendig für das Mapping von Werten zwischen Modulen

 **Rückgabewert**

Name	Typ	Beschreibung
Call	BOOL	

**6.1.2.3.3.2 Reset**

**Syntax**

```
METHOD Reset : BOOL
VAR
END_VAR
```

 **Rückgabewert**

Name	Typ	Beschreibung
Reset	BOOL	Ist TRUE, wenn abgeschlossen.

### 6.1.2.3.3.3 SetHMI

#### Syntax

```
METHOD GetHMI : BOOL
VAR_INPUT
    nContent : DINT
    pContent : PVOID
    bHMIReinit : BOOL
END_VAR

VAR
    pHMI_C[n]_[Content] : POINTER TO ST_HMI_C[n]_[Content];
END_VAR
```

#### Eingänge

Name	Typ	Beschreibung
nContent	DINT	HMI Content Index
pContent	PVOID	Pointer zur HMI Content Struktur
bHMIReinit	BOOL	Initialisieren der HMI Content Struktur

#### Rückgabewert

Name	Typ	Beschreibung
SetHMI	BOOL	Ist TRUE, wenn abgeschlossen

### 6.1.2.3.3.4 GetHMI (4.1)

#### Syntax

```
METHOD GetHMI : BOOL
VAR_INPUT
    nContent : DINT
    pContent : PVOID
END_VAR

VAR
    pHMI_C[n]_[Content] : POINTER TO ST_HMI_C[n]_[Content];
END_VAR
```

#### Eingänge

Name	Typ	Beschreibung
nContent	DINT	HMI Content Index
pContent	PVOID	Pointer zur HMI Content Struktur

#### Rückgabewert

Name	Typ	Beschreibung
GetHMI	BOOL	Ist TRUE, wenn abgeschlossen

### 6.1.2.3.4 FB\_Analysis

Im FB Analysis ist die gesamte Analyseroutine definiert. Alle konfigurierten Netzwerke mit ihren Modulen und Fehlerbehandlung werden erstellt.

#### Syntax

Definition:

```

FUNCTION_BLOCK FB_Analysis
VAR_INPUT
    ipV[n]_VirtualInputs: I_V[n]_Virtual_Input_Source;
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
END_VAR
VAR
    [network FBs]
END_VAR
    
```

**Eingänge**

Name	Typ	Beschreibung
ipV[n]_VirtualInputs	I_V[n]_Virtual_Input_Source	Daten für die Analyse von der spezifischen Datenquelle

**Ausgänge**

Name	Typ	Beschreibung
bError	BOOL	Wird TRUE, sobald eine Fehlersituation auftritt.
ipTcResult	I_TcMessage	Nachrichtenschnittstelle des TwinCAT 3 EventLoggers, die nähere Informationen über den Rückgabewert bietet.

**Methoden**

Name	Definitionsart	Beschreibung
Call [ <a href="#">▶ 89</a> ]	Local	Methode für Hintergrundkommunikation mit dem TwinCAT-Treiber. Die Methode muss zyklisch aufgerufen werden.
Reset [ <a href="#">▶ 90</a> ]	Local	Zurücksetzen der gesamten Analyse.
ResultStream [ <a href="#">▶ 90</a> ]	Local	Optional: Wenn ein Ergebnis-Stream erstellt werden muss.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

**6.1.2.3.4.1 Call**

**Syntax**

```

METHOD Call : BOOL
VAR_INPUT
    ipVirtual_Input_Source: I_V[n]_Virtual_Input_Source;
END_VAR
    
```

**Eingänge**

Name	Typ	Beschreibung
ipVirtual_Input_Source	I_V[n]_Virtual_Input_Source	Daten für die Analyse.

### Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

## 6.1.2.3.4.2 Reset

### Syntax

```
METHOD Reset : BOOL
VAR_IN_OUT
    stReset: ST_AnalysisReset;
END_VAR
```

### Eingänge

Name	Typ	Beschreibung
stReset	ST_AnalysisReset	Struktur, um zu definieren, welches Modul oder Netzwerk zurückgesetzt werden soll.

### Rückgabewert

Name	Typ	Beschreibung
Reset	BOOL	Ist TRUE, wenn abgeschlossen.

## 6.1.2.3.4.3 ResultStream

### Syntax

```
METHOD ResultStream : BOOL
VAR_INPUT
    ipResults: I_Results;
END_VAR
```

### Eingänge

Name	Typ	Beschreibung
ipResults	I_Results	Schnittstellenzeiger auf den FB Results

### Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

## 6.1.2.3.5 FB\_Results

Wenn Ergebnisse der Analyse gespeichert oder gestreamt werden müssen, verwaltet dies der FB Results und streamt die ausgewählten Variablen zum Message-Broker oder speichert die Daten in die Analytics-Binärdatei.

### Syntax

Definition:

```
FUNCTION_BLOCK FB_Results
VAR_OUTPUT
    stResults: ST_Results;
END_VAR
```

 **Ausgänge**

Name	Typ	Beschreibung
stResults	ST_Results	Ergebnisstruktur, die alle Elemente des Ergebnis-Streams enthält.

 **Methoden**

Name	Definitionsart	Beschreibung
<a href="#">Call [▶ 91]</a>	Local	Methode für Hintergrundkommunikation. Die Methode muss zyklisch aufgerufen werden.
<a href="#">AddResult [▶ 91]</a>	Local	Fügt dem Ergebnis-Stream ein Muster hinzu.
<a href="#">SendResults [▶ 92]</a>	Local	Sendet alle gepufferten Muster des Ergebnis-Streams.
<a href="#">Release [▶ 92]</a>	Local	Schließt den Stream oder die Datei des Ergebnis-Streams.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

**6.1.2.3.5.1 Call**

**Syntax**

```
METHOD Call : BOOL
VAR
END_VAR
```

 **Rückgabewert**

Name	Typ	Beschreibung
Call	BOOL	

**6.1.2.3.5.2 AddResult**

**Syntax**

```
METHOD AddResult : BOOL
VAR_INPUT
    tTimestamp: ULINT;
    stSample: ST_Results;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
tTimestamp	ULINT	Zeitstempel des Samples
stSample	ST_Results	Sample-Struktur

 **Rückgabewert**

Name	Typ	Beschreibung
AddResult	BOOL	

### 6.1.2.3.5.3 SendResults

#### Syntax

```
METHOD SendResults : BOOL
VAR
END_VAR
```

#### Rückgabewert

Name	Typ	Beschreibung
SendResults	BOOL	

### 6.1.2.3.5.4 Release

#### Syntax

```
METHOD Release : BOOL
VAR
END_VAR
```

#### Rückgabewert

Name	Typ	Beschreibung
Release	BOOL	

### 6.1.2.3.6 MAIN\_Analytics

Im Programm MAIN\_Analytics werden die DataSource FBs aufgerufen, die Reset Funktion gemanaged und ggf. die Werte mit dem HMI Dashboard gemapped. Das Programm ist einem separaten Task zugewiesen.

#### Syntax

Definition:

```
PROGRAM MAIN_Analytics
VAR_INPUT
    sCurrentStreamSystemID: GUID;
    stHistStreamInfo: REFERENCE TO ST_HMI_DataSourceHist;
    stHistRecordInfo: REFERENCE TO ST_HMI_DataSourceHistRecordInfo;
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
END_VAR
```

#### Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

### 6.1.2.3.7 MAIN\_Analytics\_C[n]

Im Programm MAIN\_Analytics\_C[n] wird der FB Analysis aufgerufen. Das Programm ist dem separaten Task zugewiesen.

Es ist auch möglich, einzelne Module, ganze Netzwerke oder alle definierten Netzwerke mit nur einem steigendem Flag zurückzusetzen. Zunächst müssen Sie die Komponente auswählen, die zurückgesetzt werden soll. Dann startet eine steigende Flanke am INPUT bReset den Reset-Vorgang.

Alle Reset-Aufrufe sind in der Aktion A\_Reset definiert.

Das Mapping der HMI Werte wird in der Aktion A\_MapToHMI vorgenommen.

**Syntax**

Definition:

```
PROGRAM MAIN_Analytics_C[n]
VAR_INPUT
    stReset: ST_AnalysisReset;
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
    nLastDataHandle: ULINT;
END_VAR
```

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

### 6.1.2.4 Codeversion 5.0

The screenshot shows the Solution Explorer for a project named 'AnalyticsSolution4'. The tree view is organized as follows:

- 1:** SYSTEM
  - License
  - Real-Time
  - Tasks
    - PlcTask
    - AnalyticsTask
    - AnalyticsTask\_C1
    - AnalyticsTask\_C2
- 2:** Routes
  - Type System
  - TcCOM Objects
    - 172.17.62.145\_Obj1 (StreamHelper)
    - 172.17.62.145\_Obj2 (StreamHelper)
- 3:** MOTION
  - PLC
    - AnalysisPLC
      - AnalysisPLC Project
        - External Types
        - References
        - DUTs
          - Analytics
            - HMI
            - Results
              - ST\_Results (STRUCT)
            - Sources
              - E\_AnalysisComponents (ENUM)
              - E\_DataSourceContext (ENUM)
              - E\_DataSourceState (ENUM)
              - E\_VirtualInput (ENUM)
              - ST\_AnalysisReset (STRUCT)
              - ST\_HistoricalDataSourceCtrl (STRUCT)
- 4:** GVLs
  - Analytics
  - AnalyticsHMI
- 5:** POU
  - Analytics
    - [Helper]
    - Analysis
      - Networks
        - 6:** N01 Network
          - Modules
            - FB\_N1\_M1\_MovingAverage1Ch (FB)
            - FB\_N1\_M2\_MinMaxAvg1Ch (FB)
            - FB\_N1\_M3\_EdgeCounter1Ch (FB)
            - FB\_N1\_M4\_ThresholdClassifier1Ch (FB)
          - FB\_N1\_Network (FB)
          - FB\_N2\_ClosedNetwork (FB)
        - FB\_Analysis (FB)
- 7:** ClosedNetworks
  - CN01 ClosedNetwork
    - Modules
      - FB\_CN1\_ClosedNetwork\_M1 (FB)
      - FB\_CN1\_ClosedNetwork\_M2 (FB)
      - FB\_CN1\_ClosedNetwork\_M3 (FB)
      - FB\_CN1\_ClosedNetwork (FB)
- 8:** Results
  - FB\_Results (FB)
  - L\_Results
- 9:** Sources
  - [Interfaces]
    - I\_InputSource
    - I\_InputSource-Historical
  - T1\_InputSource
    - FB\_T1\_C1\_InputSource-Historical (FB)
    - FB\_T1\_InputSource (FB)
- 10:** VirtualInputSources
  - [Interfaces]
    - I\_V1\_Virtual\_Input\_Source
    - I\_VirtualInputSource
  - C1\_Config\_Hist
    - FB\_V1\_C1\_Virtual\_Input\_Source (FB)
  - C2\_Config\_Live
    - FB\_V1\_C2\_Virtual\_Input\_Source (FB)
- 11:** MAIN\_Analytics (PRG)
- 12:** MAIN\_Analytics\_C1 (PRG)
- 13:** MAIN\_Analytics\_C2 (PRG)
- MAIN (PRG)
- VISUs
  - AnalyticsTask (AnalyticsTask)
  - MAIN\_Analytics
    - AnalyticsTask\_C1 (AnalyticsTask\_C1)
      - MAIN\_Analytics\_C1
    - AnalyticsTask\_C2 (AnalyticsTask\_C2)
      - MAIN\_Analytics\_C2
  - PlcTask (Plc Task)
  - AnalysisPLC Instance

#### 1: Tasks

Für die Analytics-Analyse und für jede Konfiguration einer Virtual Input Source wird ein separater Task erzeugt.

## 2: StreamHelper

Für jede Datenquellen vom Typ MQTT-Binärstrom, erstellt die Codegenerierung eine Instanz eines Stream Helper-Objekts, um die eingehenden Binärstrommuster zu verarbeiten.

## 3: DataTypes

Die Datentypen werden für die Analyse erstellt. Sie enthalten STRUCTs für die Reset-Funktion oder Ergebnisverarbeitung und ENUMs, um die verschiedenen Komponenten auszuwählen.

## 4: GVLs

Um bequem Ein- und Ausgänge der Module mit dem HMI Dashboard zu mappen, werden ausgewählte Variablen als globale Variable generiert. Zusätzlich werden die Data Source Baustein Instanzen und verschiedene Parameter als Globale Variable generiert.

## 5: Network

Alle Module werden in einem spezifischen Netzwerk sortiert, um eine bessere Übersicht und Struktur der konfigurierten Analyse zu erreichen.

## 6: Modules

Die Modul-FBs enthalten alle Eingänge und Ausgänge der konfigurierten Module aus dem Workbench-Konfigurator. Es ist auch möglich, die Module während der Laufzeit neu zu konfigurieren. Dazu müssen Sie lediglich den Parameter ändern und dann den Neukonfigurierungsprozess mit einer steigenden Flanke am INPUT bReconfigure starten.

## 7: Analysis

Im FB-Analysis ist die gesamte Analyseroutine definiert. Alle konfigurierten Netzwerke mit ihren Modulen und Fehlerbehandlung werden erstellt.

## 8: ClosedNetwork

Die ClosedNetwork-FBs werden einmal mit allen Subnetzwerken und Modulen generiert. Sie können in der Analyse mehrmalig instanziiert werden. Auf diese Weise kann der generierte Code reduziert und vereinfacht werden.

## 9: Results

Wenn Ergebnisse der Analyse gespeichert oder gestreamt werden müssen, verwaltet dies der FB-Results und streamt die ausgewählten Variablen zum Message-Broker im Binär oder Json Format, oder speichert die Daten lokal in eine Analytics-Binärdatei.

## 10: InputSource

Der FB-InputSource verwaltet den Empfang der Eingangswerte der verschiedenen Quellen. In der OUTPUT-Deklaration finden Sie alle konfigurierten Eingänge.

## 11: VirtualInputSource

Die VirtualInputSource Interfaces abstrahieren die InputSource Symbole von der Analyse.

## 12: MAIN PRGs

In dem Programm MAIN\_Analytics werden die DataSource FBs aufgerufen, die Reset Funktion verwaltet und ggf. die Werte mit dem HMI Dashboard gemapped. Das Programm ist einem separaten Task zugewiesen.

In den Programmen MAIN\_Analytics\_Vx\_Cx wird der FB-Analysis aufgerufen. Die Programme sind jeweils einem separaten Task zugewiesen.

Es ist auch möglich, einzelne Module, ganze Netzwerke oder alle definierten Netzwerke mit nur einem steigendem Flag zurückzusetzen. Zunächst müssen Sie die Komponente auswählen, die zurückgesetzt werden soll. Dann startet eine steigende Flanke am INPUT bReset den Reset-Vorgang.

Alle Reset-Aufrufe sind in der Aktion A\_Reset definiert.

### Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics Version >= 3.1.0.0

### 6.1.2.4.1 FB\_Analysis

Im FB-Analysis ist die gesamte Analyseroutine definiert. Alle konfigurierten Netzwerke mit ihren Modulen und Fehlerbehandlung werden erstellt.

#### Syntax

Definition:

```
FUNCTION_BLOCK FB_Analysis
VAR_INPUT
    nConfigurationID: INT;
    ipV[n]_VirtualInputs: I_V[n]_Virtual_Input_Source;
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
END_VAR
VAR
    [network FBs]
END_VAR
```

#### Eingänge

Name	Typ	Beschreibung
nConfigurationID	INT	Konfigurations Index
ipV[n]_VirtualInputs	I_V[n]_Virtual_Input_Source	Daten für die Analyse der spezifischen Datenquelle

#### Ausgänge

Name	Typ	Beschreibung
bError	BOOL	Wird TRUE, sobald eine Fehlersituation auftritt.
ipTcResult	I_TcMessage	Nachrichtenschnittstelle des TwinCAT 3 EventLoggers, die nähere Informationen über den Rückgabewert bietet.

#### Eigenschaften

Name	Typ	Zugriff	Beschreibung
nContext	DWORD	Get	Kontext Index

 Methoden

Name	Definitionsort	Beschreibung
<a href="#">Call</a> [ <a href="#">▶_97</a> ]	Local	Methode für Hintergrundkommunikation mit dem TwinCAT-Treiber. Die Methode muss zyklisch aufgerufen werden.
<a href="#">Reset</a> [ <a href="#">▶_97</a> ]	Local	Zurücksetzen der gesamten Analyse.
<a href="#">ResultStream</a> [ <a href="#">▶_98</a> ]	Local	Optional: Wenn ein Ergebnis-Stream erstellt werden muss.
<a href="#">SetHMIValues</a> [ <a href="#">▶_98</a> ]	Local	Methode zum Befüllen der HMI-Strukturen
<a href="#">GetHMIValues</a> [ <a href="#">▶_98</a> ]	Local	Methode zum Setzen der Parameter aus der HMI in die Analyse

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.1.2.4.1.1 Call

Syntax

```
METHOD Call : BOOL
VAR_INPUT
    ipVirtual_Input_Source: I_V[n]_Virtual_Input_Source;
END_VAR
```

 Eingänge

Name	Typ	Beschreibung
ipVirtual_Input_Source	I_V[n]_Virtual_Input_Source	Daten für die Analyse.

 Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

6.1.2.4.1.2 Reset

Syntax

```
METHOD Reset : BOOL
VAR_IN_OUT
    stReset: ST_AnalysisReset;
END_VAR
```

 Eingänge

Name	Typ	Beschreibung
stReset	ST_AnalysisReset	Struktur, um zu definieren, welches Modul oder Netzwerk zurückgesetzt werden soll.

### Rückgabewert

Name	Typ	Beschreibung
Reset	BOOL	Ist TRUE, wenn abgeschlossen.

## 6.1.2.4.1.3 ResultStream

### Syntax

```
METHOD ResultStream : BOOL
VAR_INPUT
    ipResults: I_Results;
END_VAR
```

### Eingänge

Name	Typ	Beschreibung
ipResults	I_Results	Schnittstellenzeiger auf den FB-Results

### Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

## 6.1.2.4.1.4 SetHMIValues

### Syntax

```
METHOD SetHMIValues : BOOL
VAR_INPUT
    pHMI_N[n]_[Network1] : POINTER TO ST_HMI_N[n]_[Network1];
END_VAR
```

### Eingänge

Name	Typ	Beschreibung
pHMI_N[n]_[Network1]	POINTER TO ST_HMI_N[n]_[Network1]	Zeiger zur globalen HMI-Struktur

### Rückgabewert

Name	Typ	Beschreibung
SetHMIValues	BOOL	Ist TRUE, wenn abgeschlossen

## 6.1.2.4.1.5 GetHMIValues

### Syntax

```
METHOD GetHMIValues : BOOL
VAR_INPUT
    pHMI_N[n]_[Network1] : POINTER TO ST_HMI_N[n]_[Network1];
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
pHMI_N[n]_[Network1]	POINTER TO ST_HMI_N[n]_[Network1]	Zeiger zur globalen HMI-Struktur

 **Rückgabewert**

Name	Typ	Beschreibung
SetHMIValues	BOOL	Ist TRUE, wenn abgeschlossen

**6.1.2.4.2 FB\_Network**

Alle Module werden in einem spezifischen Netzwerk sortiert, um eine bessere Übersicht und Struktur der konfigurierten Analyse zu erreichen.

**Syntax**

Definition:

```
FUNCTION_BLOCK FB_N[n]_[Network1]
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
END_VAR
VAR
    [module FBs]
END_VAR
```

 **Ausgänge**

Name	Typ	Beschreibung
bError	BOOL	Wird TRUE, sobald eine Fehlersituation auftritt.
ipTcResult	I_TcMessage	Nachrichtenschnittstelle des TwinCAT 3 EventLoggers, die nähere Informationen über den Rückgabewert bietet.

 **Methoden**

Name	Definitionsart	Beschreibung
<a href="#">Call [▶ 74]</a>	Local	Methode für Hintergrundkommunikation. Die Methode muss zyklisch aufgerufen werden.
<a href="#">Reset [▶ 74]</a>	Local	Zurücksetzen des Netzwerks mit allen Submodulen.
<a href="#">ValueMapping [▶ 100]</a>	Local	Zuordnung der Eingangswerte zu den verschiedenen Moduleingängen.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

### 6.1.2.4.2.1 Call

#### Syntax

```
METHOD Call : BOOL
VAR_INPUT
    pAnalysis: POINTER TO FB_Analysis;
END_VAR
```

#### Eingänge

Name	Typ	Beschreibung
pAnalysis	FB_Analysis	Instanz des Analyse FBs.

#### Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

### 6.1.2.4.2.2 Reset

#### Syntax

```
METHOD Reset : BOOL
VAR
END_VAR
```

#### Rückgabewert

Name	Typ	Beschreibung
Reset	BOOL	Ist TRUE, wenn abgeschlossen.

### 6.1.2.4.2.3 ValueMapping

#### Syntax

```
METHOD ValueMapping : BOOL
VAR_INPUT
    pAnalysis : POINTER TO FB_Analysis;
END_VAR
```

#### Eingänge

Name	Typ	Beschreibung
pAnalysis	FB_Analysis	Instanz des Analyse FBs

#### Rückgabewert

Name	Typ	Beschreibung
ValueMapping	BOOL	

### 6.1.2.4.3 FB\_Module

Die Modul-FBs enthalten alle Eingänge und Ausgänge der konfigurierten Module aus dem Workbench-Konfigurator. Es ist auch möglich, die Module während der Laufzeit neu zu konfigurieren. Dazu müssen Sie lediglich den Parameter ändern und dann den Neukonfigurierungsprozess mit einer steigenden Flanke am INPUT bReconfigure starten.

**Syntax**

**Definition:**

```
FUNCTION_BLOCK FB_N[n]_M[n]_[Module]
VAR_INPUT
    [module inputs]
END_VAR
VAR_INPUT PERSISTENT
    [module persistent parameter inputs]
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
    [module outputs]
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
Module inputs		Eingänge des ausgewählten Moduls.

 **Ausgänge**

Name	Typ	Beschreibung
bError	BOOL	Wird TRUE, sobald eine Fehlersituation auftritt.
ipTcResult	I_TcMessage	Nachrichtenschnittstelle des TwinCAT 3 EventLoggers, die nähere Informationen über den Rückgabewert bietet.
Module outputs		Ausgänge des ausgewählten Moduls.

 **Eigenschaften**

Name	Typ	Zugriff	Beschreibung
nContext	DWORD	Get	Kontext Index

 **Methoden**

Name	Definitionsart	Beschreibung
<a href="#">Call [▶_102]</a>	Local	Methode für Hintergrundkommunikation. Die Methode muss zyklisch aufgerufen werden.
<a href="#">Reset [▶_102]</a>	Local	Zurücksetzen des Moduls.
<a href="#">SetHMI [▶_102]</a>	Local	Optional: Setzt die Ein-/Ausgänge auf die globalen HMI-Strukturen.
<a href="#">GetHMI [▶_103]</a>	Local	Optional: Setzt die Eingänge der globalen HMI-Strukturen auf die Eingänge des Moduls

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

### 6.1.2.4.3.1 Call

#### Syntax

```
METHOD Call : BOOL
VAR_INPUT
    ipVirtual_Input_Source : I_V[n]_Virtual_Input_Source;
    [ipValueMapping_M2M : I_ValueMapping_M2M;]
END_VAR
```

#### Eingänge

Name	Typ	Beschreibung
ipVirtual_Input_Source	I_V[n]_Virtual_Input_Source	Daten für die Analyse
ipValueMapping_M2M	I_ValueMapping_M2M	Optional: Notwendig für das Mapping von Werten zwischen Modulen

#### Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

### 6.1.2.4.3.2 Reset

#### Syntax

```
METHOD Reset : BOOL
VAR
END_VAR
```

#### Rückgabewert

Name	Typ	Beschreibung
Reset	BOOL	Ist TRUE, wenn abgeschlossen.

### 6.1.2.4.3.3 SetHMI

#### Syntax

```
METHOD SetHMI : BOOL
VAR_INPUT
    nContent : DINT
    pContent : PVOID
    bHMIReinit : BOOL
END_VAR
VAR
    pHMI_C[n]_[Content] : POINTER TO ST_HMI_C[n]_[Content];
END_VAR
```

#### Eingänge

Name	Typ	Beschreibung
nContent	DINT	HMI Content Index
pContent	PVOID	Pointer zur HMI Content Struktur
bHMIReinit	BOOL	Initialisieren der HMI Content Struktur

 **Rückgabewert**

Name	Typ	Beschreibung
SetHMI	BOOL	Ist TRUE, wenn abgeschlossen

**6.1.2.4.3.4 GetHMI**

**Syntax**

```
METHOD GetHMI : BOOL
VAR_INPUT
    nContent : DINT
    pContent : PVOID
END_VAR
VAR
    pHMI_C[n][Content] : POINTER TO ST_HMI_C[n][Content];
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
nContent	DINT	HMI Content Index
pContent	PVOID	Pointer zur HMI Content Struktur

 **Rückgabewert**

Name	Typ	Beschreibung
GetHMI	BOOL	Ist TRUE, wenn abgeschlossen

**6.1.2.4.4 FB\_InputSource**

Der FB-DataSource verwaltet den Empfang der Eingangswerte der verschiedenen Quellen. In der OUTPUT-Deklaration finden Sie alle konfigurierten Eingänge.

**Syntax**

Definition:

```
FUNCTION_BLOCK FB_T[n]_InputSource IMPLEMENTS I_InputSource
VAR
END_VAR
```

 **Schnittstellen**

Typ	Beschreibung
I_InputSource	Schnittstelle zur Kommunikation mit einer Datenquelle

### Methoden

Name	Definitionsort	Beschreibung
<a href="#">Call</a> [ <a href="#">▶ 104</a> ]	Local	Methode für Hintergrundkommunikation mit dem TwinCAT-Treiber. Die Methode muss zyklisch aufgerufen werden.
<a href="#">GetData</a> [ <a href="#">▶ 104</a> ]	Local	Methode zum Abrufen der Daten des spezifizierten Elements.
<a href="#">NewDataAvailable</a> [ <a href="#">▶ 105</a> ]	Local	Methode zum Prüfen ob neue Daten verfügbar sind.
<a href="#">AddClient</a> [ <a href="#">▶ 105</a> ]	Local	Methode zum Hinzufügen von Daten Clients
<a href="#">ClientDone</a> [ <a href="#">▶ 106</a> ]	Local	Methode zum Signalisieren das der Client alle Daten empfangen hat.

### Eigenschaften

Name	Typ	Zugriff	Beschreibung
bReadNewData	BOOL	Get	.
eDataState	E_DataSourceState	Get	
nDataHandle	ULINT	Get	
nElements	UDINT	Get	
nMaxOversamplingFactor	UDINT	Get	

### Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

#### 6.1.2.4.4.1 Call

##### Syntax

```
METHOD Call : BOOL
```

### Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

#### 6.1.2.4.4.2 GetData

##### Syntax

```
METHOD GetData : BOOL
VAR_INPUT
  nElement      : UDINT;
  pInputs       : PVOID;
  nInputsSize   : UDINT
END_VAR
VAR_OUTPUT
  nTimestamp    : ULINT;
  nContext      : DWORD;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
nElement	UDINT	Element-ID, um das spezifische Beispiel zu erhalten
pInputs	PVOID	Pointer zur Datenstruktur
nInputsSize	UDINT	Größe der Datenstruktur

 **Ausgänge**

Name	Typ	Beschreibung
nTimestamp	ULINT	Daten Zeitstempel
nContext	DWORD	Daten Kontext

 **Rückgabewert**

Name	Typ	Beschreibung
GetData	BOOL	Ist TRUE, wenn ein neues Element ausgewählt ist

**6.1.2.4.4.3      NewDataAvailable**

**Syntax**

```
METHOD NewDataAvailable : BOOL
VAR_INPUT
    nLastDataHandle : ULINT;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
nLastDataHandle	ULINT	Handle vom zuletzt abgeholten Datenpacket

 **Rückgabewert**

Name	Typ	Beschreibung
NewDataAvailable	BOOL	Ist TRUE, wenn neue Daten verfügbar sind

**6.1.2.4.4.4      AddClient**

**Syntax**

```
METHOD AddClient : BOOL
VAR_OUTPUT
    nClientID : DWORD;
END_VAR
```

 **Ausgänge**

Name	Typ	Beschreibung
nClientID	LWORD	Client-ID

### 6.1.2.4.4.5 ClientDone

#### Syntax

```
METHOD ClientDone : BOOL
VAR_OUTPUT
    nClientID : DWORD;
END_VAR
```

#### Ausgänge

Name	Typ	Beschreibung
nClientID	LWORD	Client-ID

### 6.1.2.4.5 FB\_VirtualInputSource

Der FB VirtualInputSource abstrahiert die InputSources für die verschiedenen Analyse Konfigurationen. Die in der Analytics Workbench konfigurierten virtuellen Eingänge werden hier gesetzt.

#### Syntax

Definition:

```
FUNCTION_BLOCK FB_V[n]_C[n]_VirtualInputSource IMPLEMENTS I_VirtualInputSource,
I_V[n]_VirtualInputSource
VAR
END_VAR
```

#### Schnittstellen

Typ	Beschreibung
I_VirtualInputSource	Schnittstelle zur Kommunikation mit einer Datenquelle
I_V[n]_VirtualInputSource	Schnittstelle die alle definierten virtuellen Eingänge bereitstellt

#### Methoden

Name	Definitionsart	Beschreibung
<a href="#">SourceSync [► 107]</a>	Local	Methode zum Synchronisieren von mehreren FB-Instanzen
<a href="#">NextData [► 107]</a>	Local	Methode zum entgegenzunehmen des nächsten Datensatzes
<a href="#">Done [► 107]</a>	Local	Methode zum Signalisieren, dass das gesamte Datenpaket fertig bearbeitet wurde.

#### Eigenschaften

Name	Typ	Zugriff	Beschreibung
bEndOfData	BOOL	Get	Signalisiert das Ende des Datenpakets
dtTimestamp	DCTIMESTRUC T	Get	Zeitstempel des aktuell entgegengenommenen Datensatz
nDataHandle	ULINT	Get	Daten-Handle
nContext	DWORD	Get	Daten-Kontext
tTimestamp	ULINT	Get	Zeitstempel des aktuell entgegengenommenen Datensatzes

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

**6.1.2.4.5.1 SourceSync**

**Syntax**

```
METHOD SourceSync : BOOL
```

**6.1.2.4.5.2 NextData**

**Syntax**

```
METHOD NextData : BOOL
VAR_INPUT
    nMaxOversampling : UDINT;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
nMaxOversampling	UDINT	Gibt den maximalen Oversampling-Wert an

 **Rückgabewert**

Name	Typ	Beschreibung
NextData	BOOL	Ist TRUE, wenn ein neues Element ausgewählt ist

**6.1.2.4.5.3 Done**

**Syntax**

```
METHOD Done : BOOL
VAR_INPUT
END_VAR
```

**6.1.2.4.6 FB\_Results**

Wenn Ergebnisse der Analyse gespeichert oder gestreamt werden müssen, verwaltet dies der FB-Results und streamt die ausgewählten Variablen zum Message-Broker oder speichert die Daten in die Analytics-Binärdatei.

**Syntax**

Definition:

```
FUNCTION_BLOCK FB_Results
VAR_OUTPUT
    nTimestamp: ULINT;
    stResults: ST_Results;
END_VAR
```

 **Ausgänge**

Name	Typ	Beschreibung
nTimestamp	ULINT	Zugehöriger Zeitstempel der Ergebnisstrukturdaten
stResults	ST_Results	Ergebnisstruktur, die alle Elemente des Ergebnis-Streams enthält.

 **Eigenschaften**

Name	Typ	Zugriff	Beschreibung
bInitialized	BOOL	Get	Gibt an ob der Bausteine ordnungsgemäß initialisiert wurde
nMaxSamples	INT	Get	Maximale Anzahl an gepufferten Ergebnissen
nResultCount	INT	Get	Aktuelle Anzahl an gepufferten Ergebnissen

 **Methoden**

Name	Definitionsart	Beschreibung
<a href="#">Call [▶_108]</a>	Local	Methode für Hintergrundkommunikation. Die Methode muss zyklisch aufgerufen werden.
<a href="#">AddResult [▶_108]</a>	Local	Fügt dem Ergebnis-Stream ein Muster hinzu.
<a href="#">SendResults [▶_109]</a>	Local	Sendet alle gepufferten Muster des Ergebnis-Streams.
<a href="#">Release [▶_109]</a>	Local	Schließt den Stream oder die Datei des Ergebnis-Streams.

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

**6.1.2.4.6.1 Call****Syntax**

```
METHOD Call : BOOL
VAR
END_VAR
```

 **Rückgabewert**

Name	Typ	Beschreibung
Call	BOOL	

**6.1.2.4.6.2 AddResult****Syntax**

```
METHOD AddResult : BOOL
VAR_INPUT
    tTimestamp: ULINT;
    stSample: ST_Results;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
tTimestamp	ULINT	Zeitstempel des Samples
stSample	ST_Results	Sample-Struktur

 Rückgabewert

Name	Typ	Beschreibung
AddResult	BOOL	

**6.1.2.4.6.3 SendResults**

**Syntax**

```
METHOD SendResults : BOOL
VAR
END_VAR
```

 Rückgabewert

Name	Typ	Beschreibung
SendResults	BOOL	

**6.1.2.4.6.4 Release**

**Syntax**

```
METHOD Release : BOOL
VAR
END_VAR
```

 Rückgabewert

Name	Typ	Beschreibung
Release	BOOL	

**6.1.2.4.7 MAIN\_Analytics**

Im Programm MAIN\_Analytics werden die InputSource FBs aufgerufen, die Reset Funktion gemanaged und ggf. die Werte mit dem HMI Dashboard gemapped. Das Programm ist einem separaten Task zugewiesen.

**Syntax**

Definition:

```
PROGRAM MAIN_Analytics
VAR
END_VAR
```

**Voraussetzungen**

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

**6.1.2.4.8 MAIN\_Analytics\_C[n]**

Im Programm MAIN\_Analytics\_C[n] wird der FB-Analysis aufgerufen. Das Programm ist dem separaten Task zugewiesen.

Es ist auch möglich, einzelne Module, ganze Netzwerke oder alle definierten Netzwerke mit nur einem steigendem Flag zurückzusetzen. Zunächst müssen Sie die Komponente auswählen, die zurückgesetzt werden soll. Dann startet eine steigende Flanke am INPUT bReset den Reset-Vorgang.

Alle Reset-Aufrufe sind in der Aktion A\_Reset definiert.

Das Handling der „InputSources“ wird in der Aktion A\_InputSources durchgeführt.

Das Mapping der HMI-Werte wird in der Aktion A\_MapToHMI vorgenommen.

## Syntax

Definition:

```
PROGRAM MAIN_Analytics_C[n]
VAR_INPUT
    stReset: ST_AnalysisReset;
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
    nAnalysisResultsTimestamp: ULINT;
    stAnalysisResults: ST_Results;
END_VAR
```

## Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

## 6.2 HMI One-Click Dashboard

Es ist möglich, ein HMI Dashboard mit HMI Controls für alle Module und Parameter, die im TwinCAT Analytics Workbench Configurator konfiguriert sind, automatisch zu generieren. Das HMI Dashboard basiert auf der [TwinCAT HMI](#) und visualisiert die SPS-Daten aus dem [Runtime-Deployment](#) [► 44].

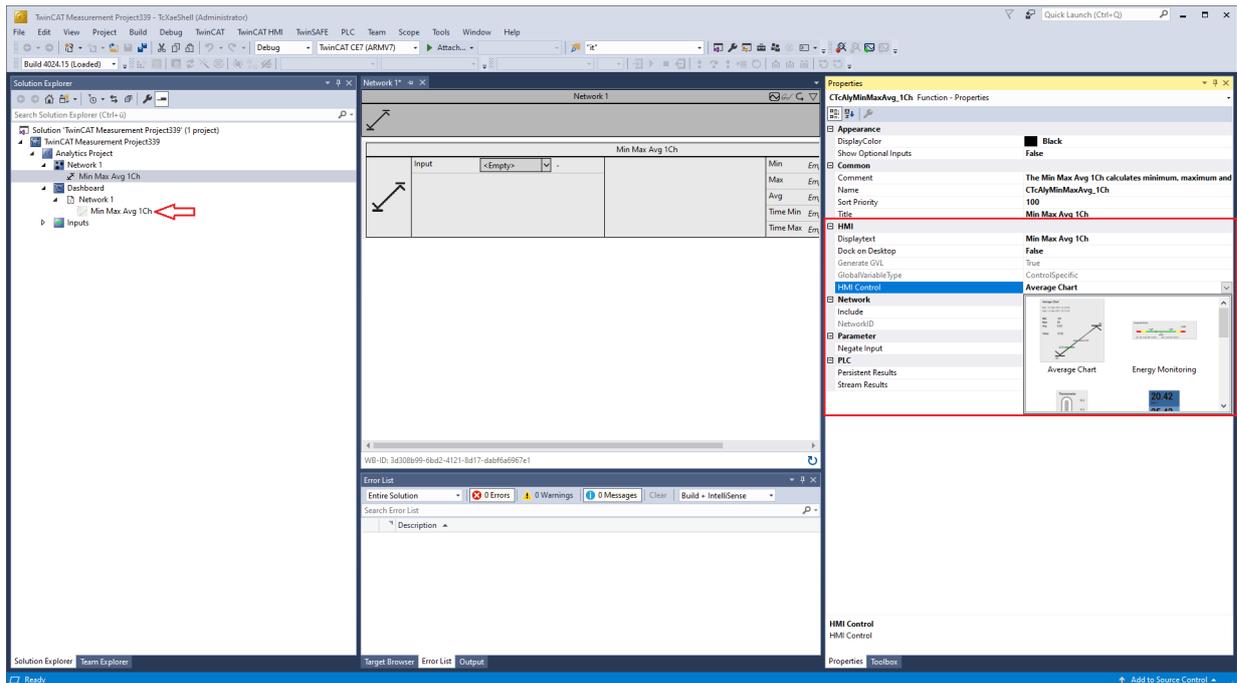


Das automatisch generierte One-Click Dashboard ist nur mit der neuen HMI Version 1.12 verfügbar. Um die Analytics HMI Controls zu verwenden, ist eine Analytics Runtime-Lizenz erforderlich.

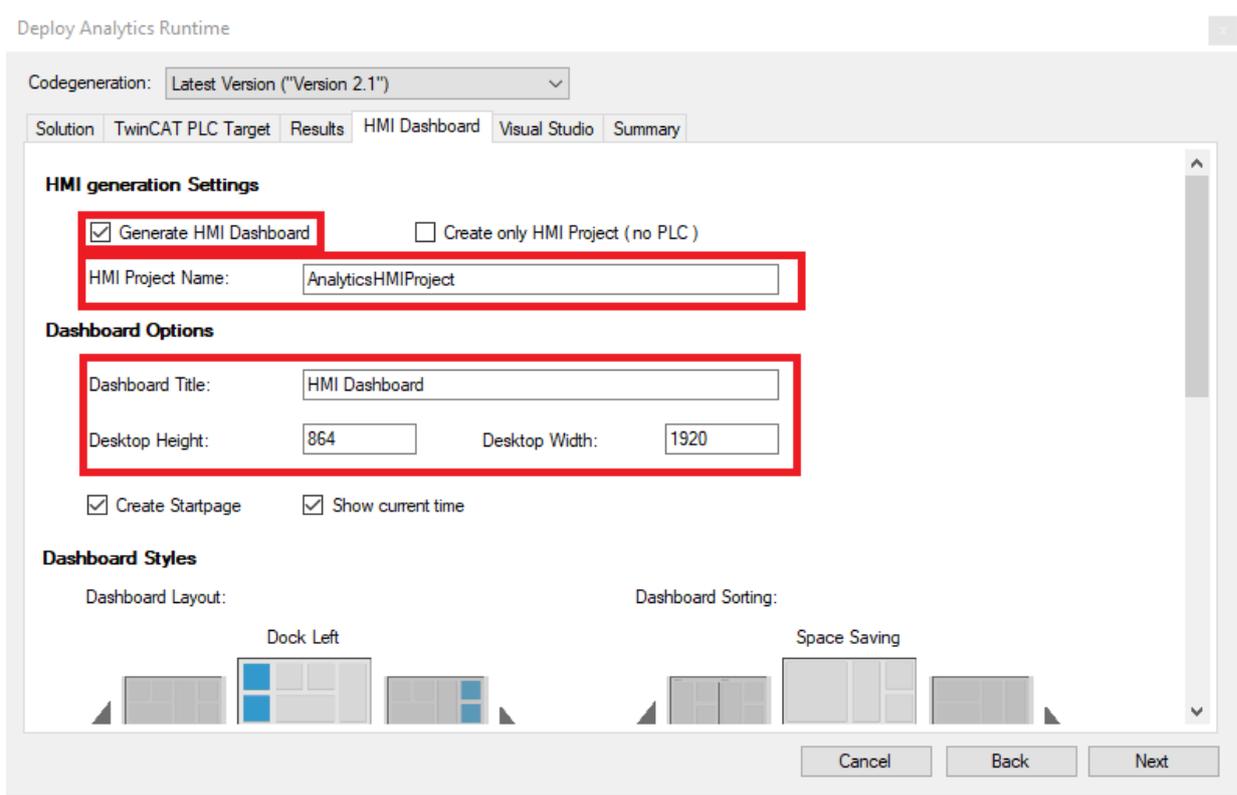


- ✓ Nach der Konfiguration Ihres Analytics Workbench-Projekts kann für jeden Algorithmus ein HMI Control ausgewählt werden.
1. Öffnen Sie das Fenster **Properties** des Moduls und wählen Sie ein **HMI Control** aus der DropDown-Liste aus. Sie können den Anzeigetext für den Titel im HMI Dashboard ändern (Displaytext). Ebenfalls können Sie auswählen, ob das Control an die Startseite andockt werden soll (Dock on Desktop). Im

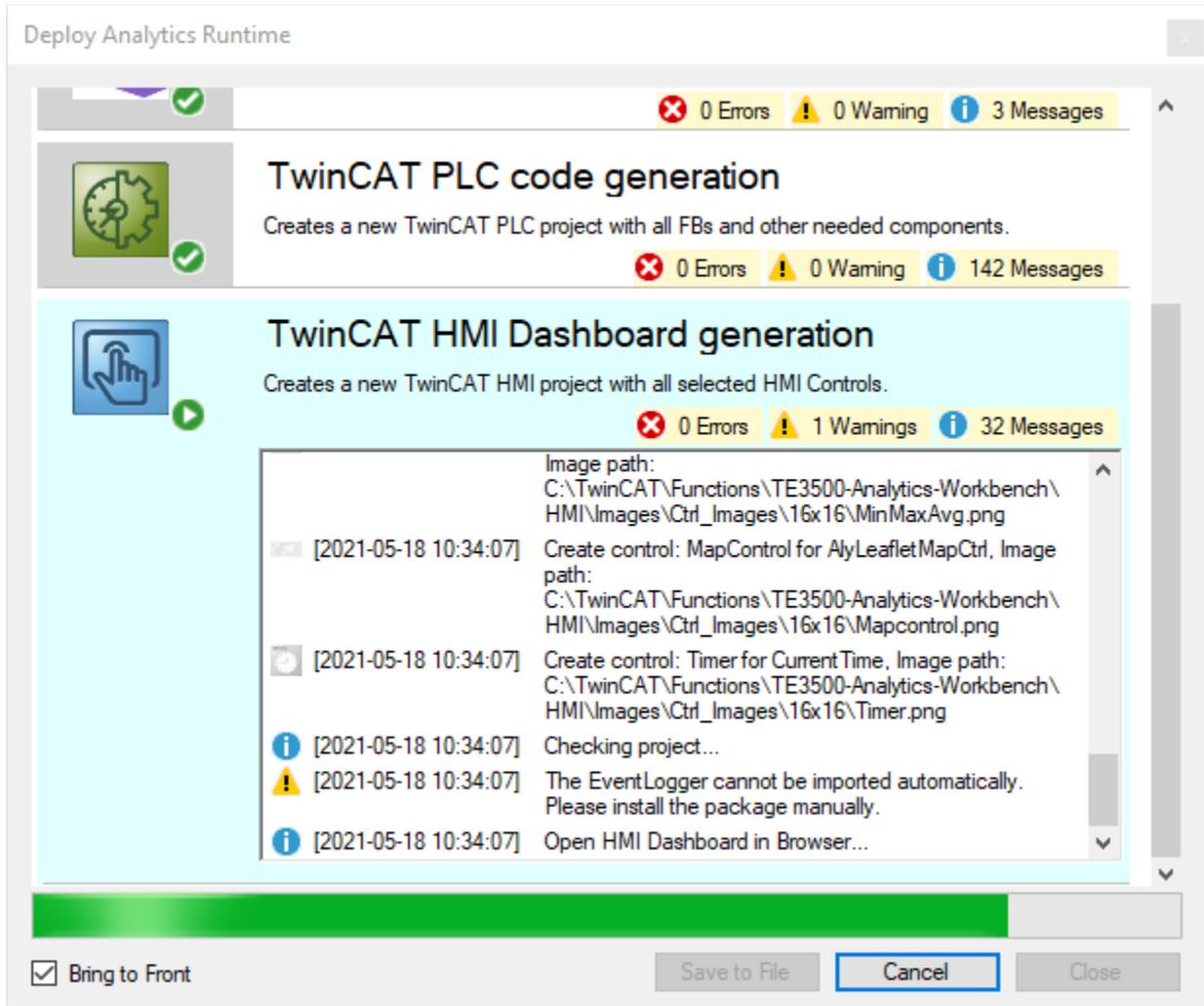
**Solution Explorer** werden alle zu generierenden Controls unter dem **Dashboard Aufbau und Inhalte im Analytics Projekt** verwalten |▶ 113| gespeichert



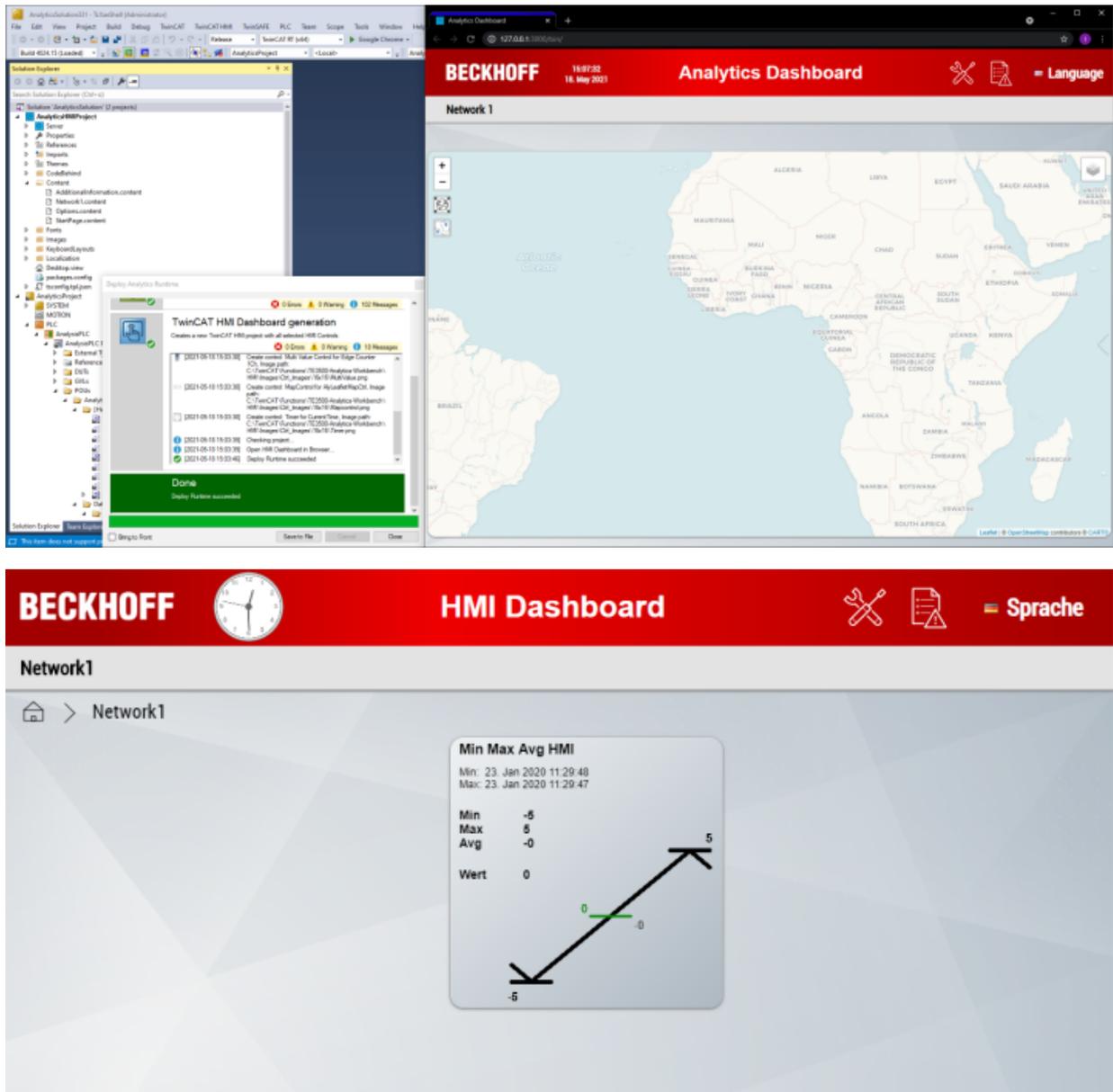
2. Nach Abschluss der Konfiguration klicken Sie auf den Befehl **Deploy Analytics Runtime** im Kontextmenü. Der Deploy-Assistent startet und es ist möglich, die gesamte benötigte Konfiguration für den Einsatz schrittweise einzurichten.
  - ⇒ Auf dem Karteireiter **HMI Dashboard** können Sie Ihr HMI Dashboard konfigurieren.
3. Aktivieren Sie die Checkbox **Generate HMI Dashboard**. Es ist auch möglich, nur ein HMI-Projekt ohne eine SPS zu erstellen. Des Weiteren können Sie einen **HMI Project Name** für das Dashboard vergeben und einen Dashboard Title sowie die Desktop Height und Desktop Width festlegen, um ein maßgeschneidertes Dashboard zu generieren, das zu Ihren Bedürfnissen passt. Die übrigen Konfigurationen werden unter **Dashboard-Konfiguration** |▶ 145| erläutert.



- ⇒ Wie üblich zeigt der letzte Karteireiter Ihnen alle Einstellungen, die Sie für die Generierung vorgenommen haben.
- 4. Nun können Sie den Generierungsprozess starten, indem Sie auf **Deploy** klicken.
- ⇒ Die HMI-Generierung beginnt direkt nach dem Runtime-Deployment (falls ausgewählt). Im Übersichtsfenster wird jeder Schritt während des Generierungsprozesses auch für die Generierung des HMI Dashboards angezeigt.

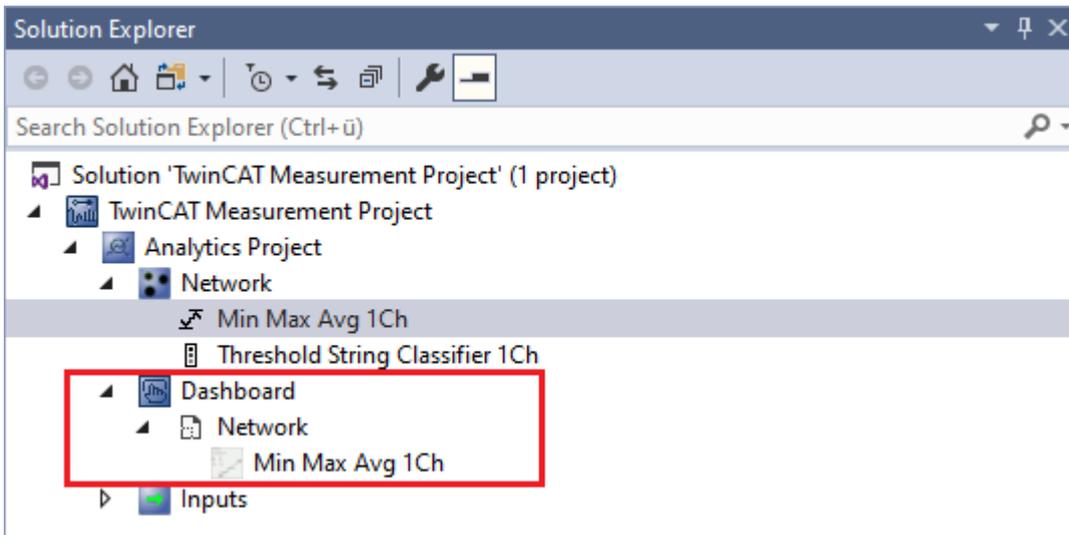


⇒ Das Dashboard wird automatisch in Ihrem Standard-Browser geöffnet.

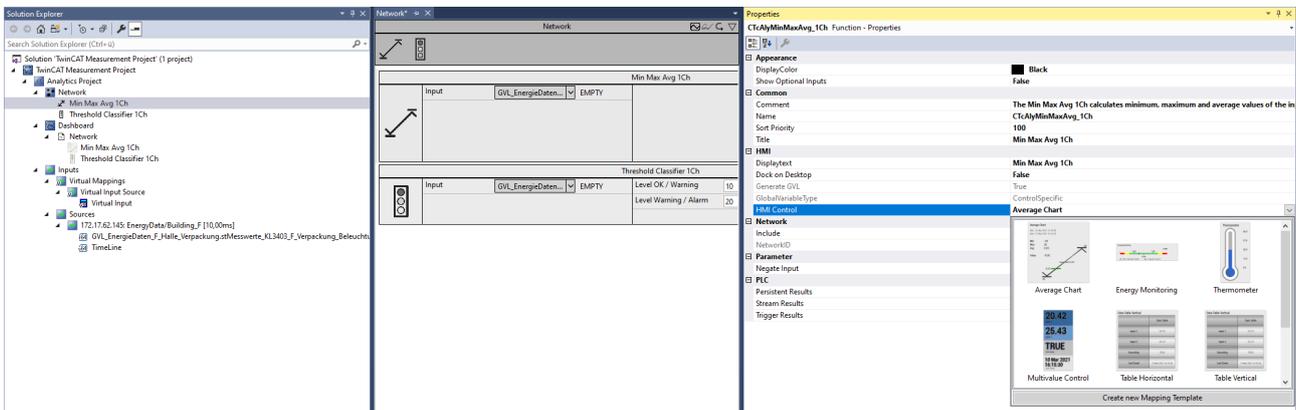


### 6.2.1 Dashboard Aufbau und Inhalte im Analytics Projekt verwalten

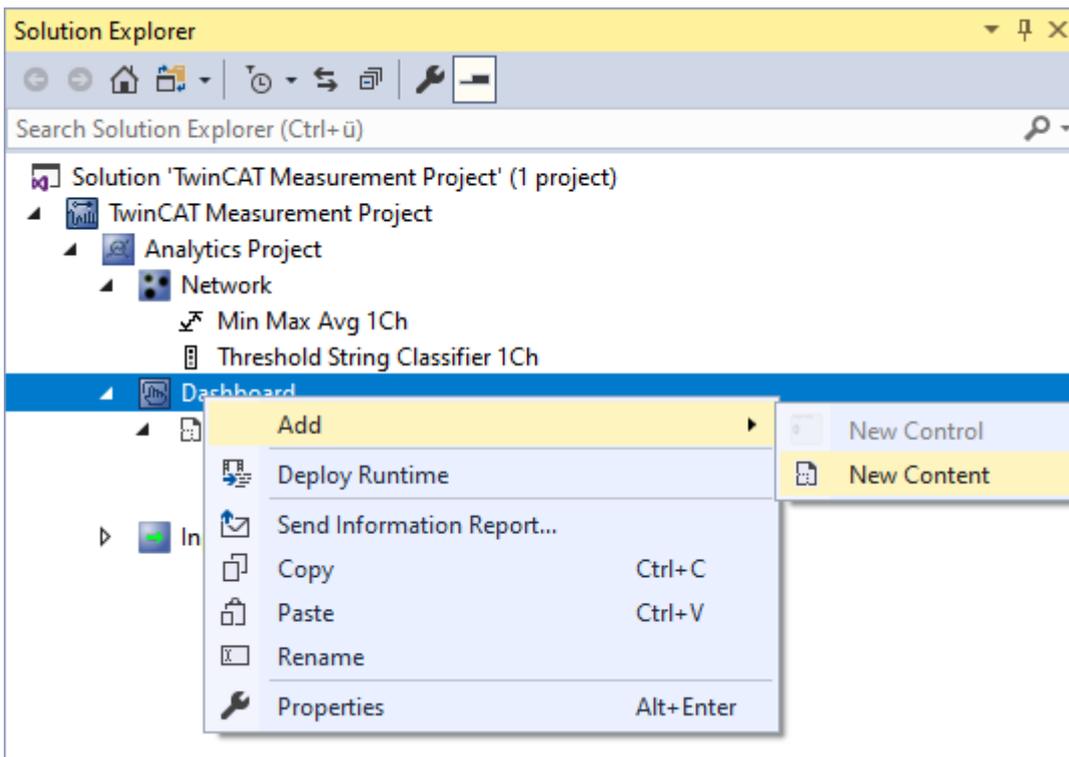
Zu einem Analytics Projekt liegt eine Konfiguration von HMI Contents (Seiten) und HMI Controls (Anzeigeelemente) vor, die bei der Dashboard Generierung erstellt werden. Diese Konfiguration ist über den **Dashboard-Knoten** im Solution Explorer ersichtlich und veränderbar. Sie können die Contents und Controls zu jeder Zeit umbenennen, in andere Contents via Drag & Drop verschieben, Kopieren (**Ctrl-C**, **Ctrl-V**) und löschen. Ein Control kann ebenfalls über einen Doppelklick editiert werden.



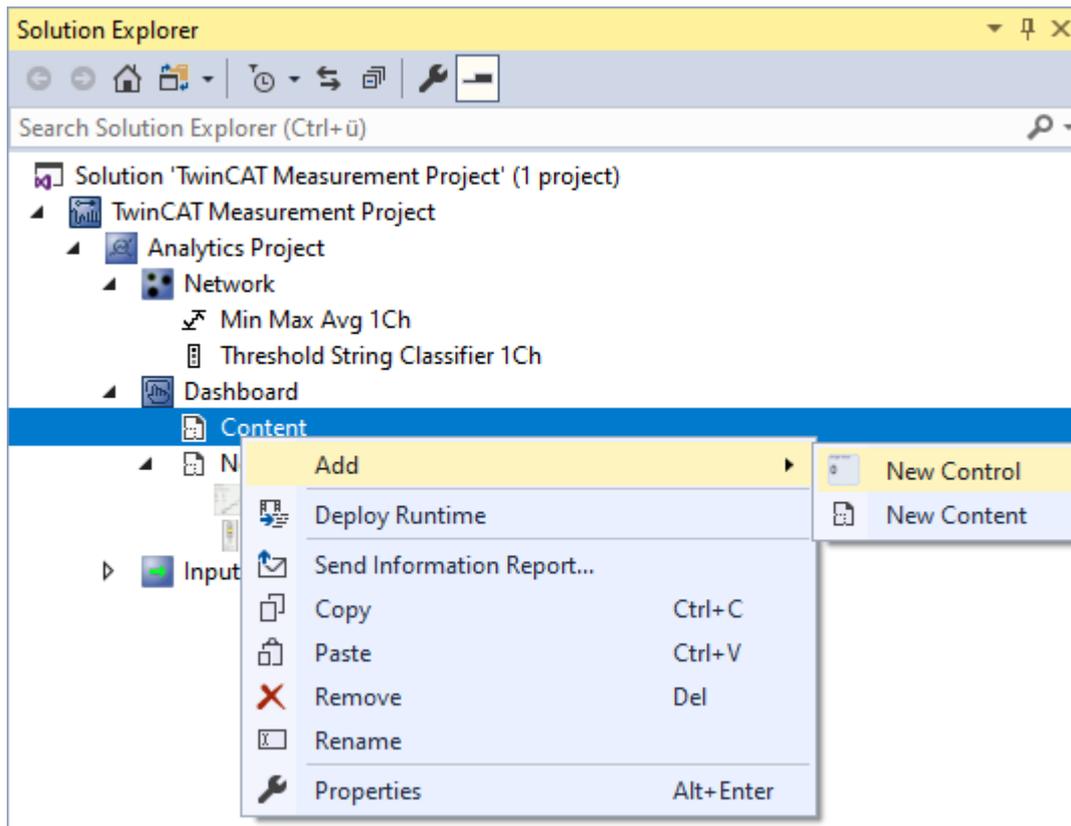
Jedes Analytics Modul besitzt bestehende Control-Mappings, welche über das Fenster **Properties** auswählbar sind (diese können Sie auch über „Create new Mapping Template“ bzw. über den Angepasste und eigene Controls verwenden [► 123] (Punkt 2) selbst erstellen/bearbeiten). Sobald ein Control für ein Modul ausgewählt wird, wird dieses unter dem Knoten **Dashboard** gelistet.



Auf dem Knoten **Dashboard** und dem Knoten **Content** können Sie via Rechtsklick neue Contents anlegen.



Ebenso können Sie **neue Controls** mit einen Rechtsklick auf einen Knoten **Content** hinzufügen.



Es öffnet sich der **Analytics Dashboard Wizard**. In diesem Wizard werden Sie Schritt für Schritt durch die Konfiguration zum Hinzufügen eines Controls geleitet.

### ● Analytics Dashboard Wizard

**I** Machen Sie sich mit Wizard vertraut, indem Sie ein Control hinzufügen (Control über das **Properties** Fenster vom Modul auswählen). Anschließend können Sie die gesamte Konfiguration im Wizard nachsehen, indem Sie einen Doppelklick auf das Control machen. Dort können Sie die einzelnen Konfigurationsschritte nachvollziehen.

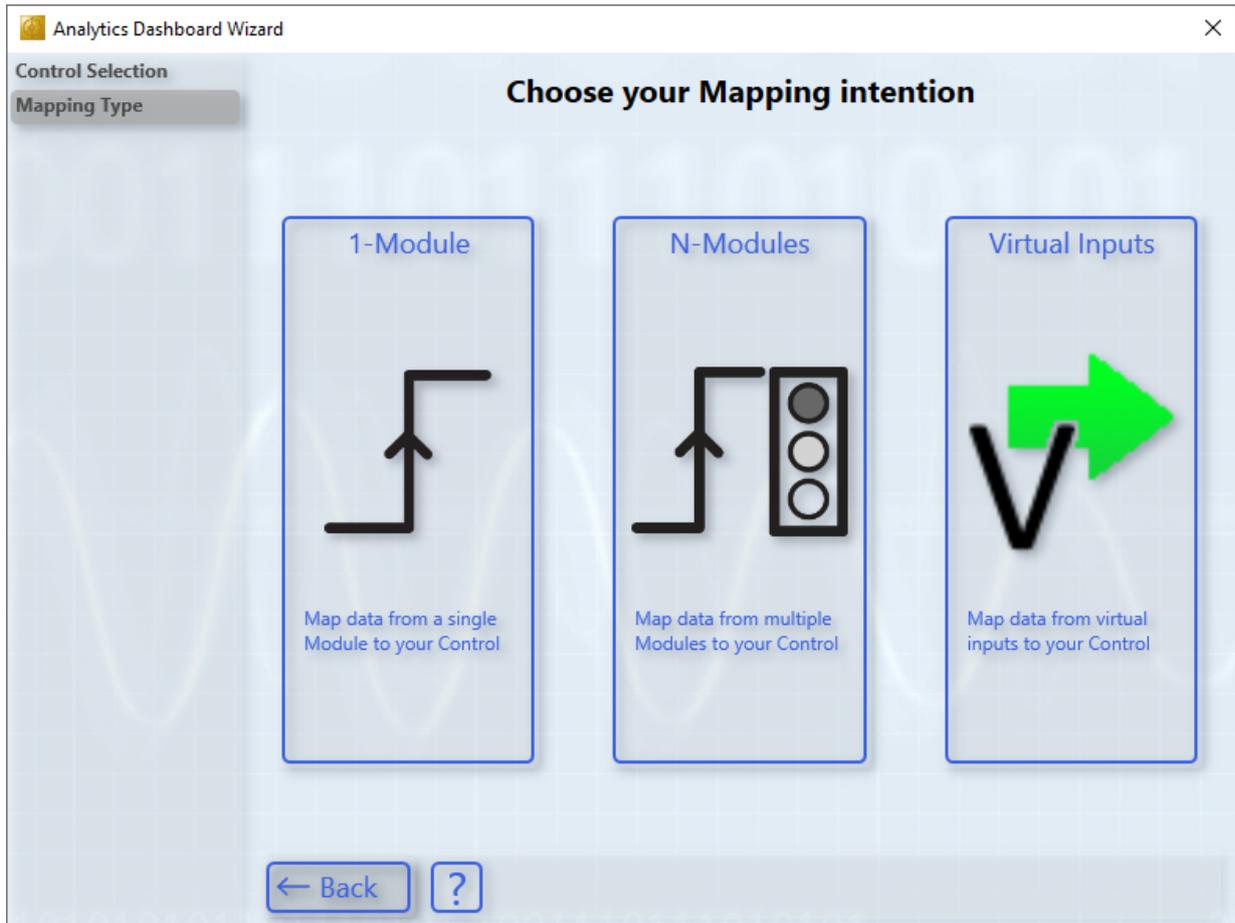
### Analytics Dashboard Wizard

Wählen Sie ein Control aus. Auf der linken Seite sind alle verfügbaren Controls gelistet. Standardmäßig werden nur Controls gelistet, die Algorithmus unspezifisch sind. Über das Abhaken der Checkbox **Show only default Controls** sind alle Controls verfügbar.

1. Wählen Sie **Single Value**, um einen einzelnen Wert im Dashboard anzuzeigen. Drücken Sie auf den Button **Next**, um fortzufahren.



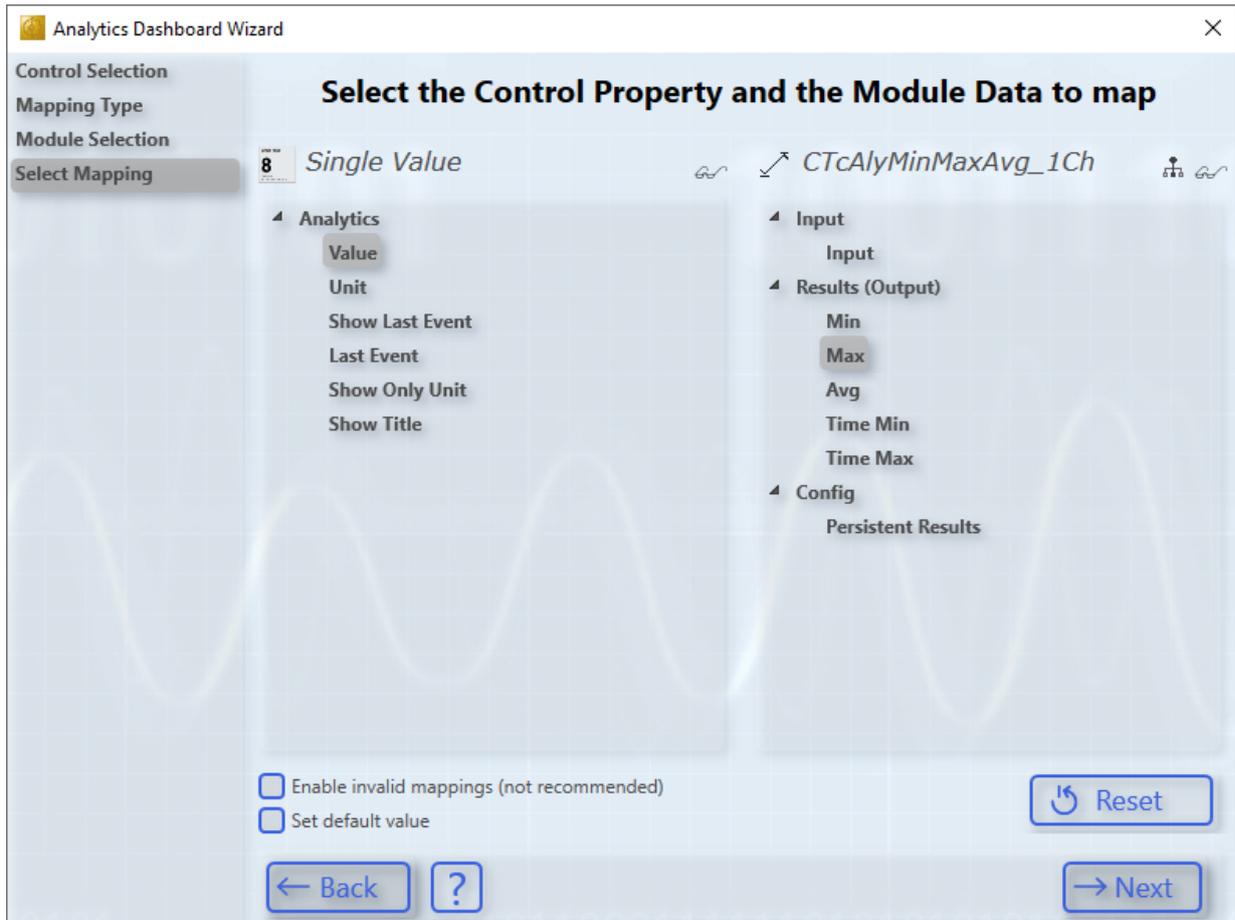
2. Sie können nun Daten von einem Modul (**1-Module**), mehreren Modulen (**N-Modules**) oder von Virtuellen Inputs (**Virtual Inputs**) verknüpfen. Wählen Sie für dieses Beispiel **1-Module** aus.



- Es werden alle Module aus dem Analytics Projekt aufgelistet. Selektieren Sie das Modul, von dem Sie die Daten darstellen möchten. Auf der rechten Seite befindet sich eine Vorschau zu dem selektierten Modul.



- Hier erfolgt die Verknüpfung der ersten Control Eigenschaft mit einer Modul Variable. Wählen Sie dazu den anzuzeigenden Wert **Value** mit der Variable **Max**.



5. Auf dieser Übersichtsseite werden alle bestehenden Verknüpfungen aufgelistet. Fügen Sie mit **Add** eine weitere Verknüpfung hinzu.

**Analytics Dashboard Wizard** ×

**Mapping Overview**

*Single Value* *CTcAlyMinMaxAvg1Ch*

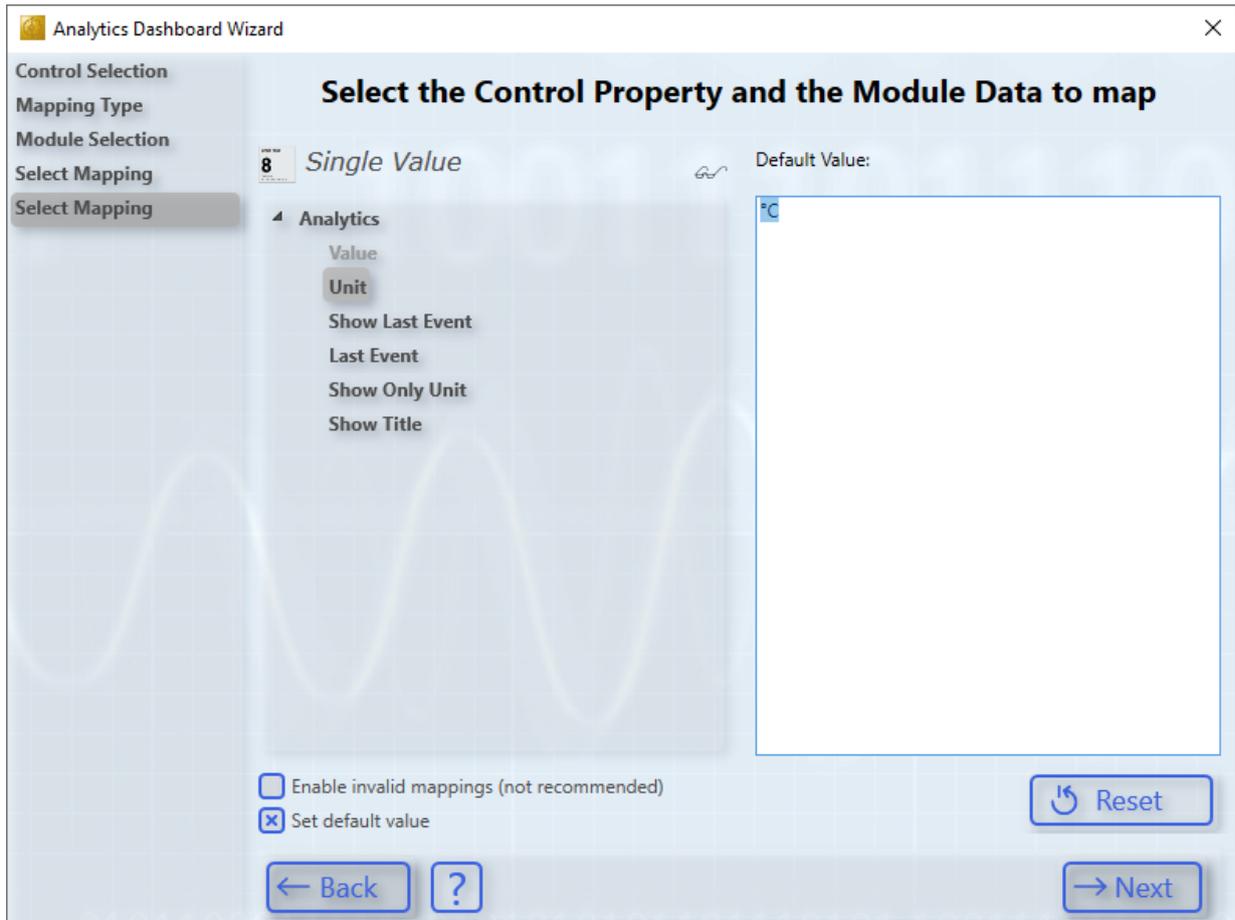
Control Property	Data	
Value	Max	-

**+ Add**

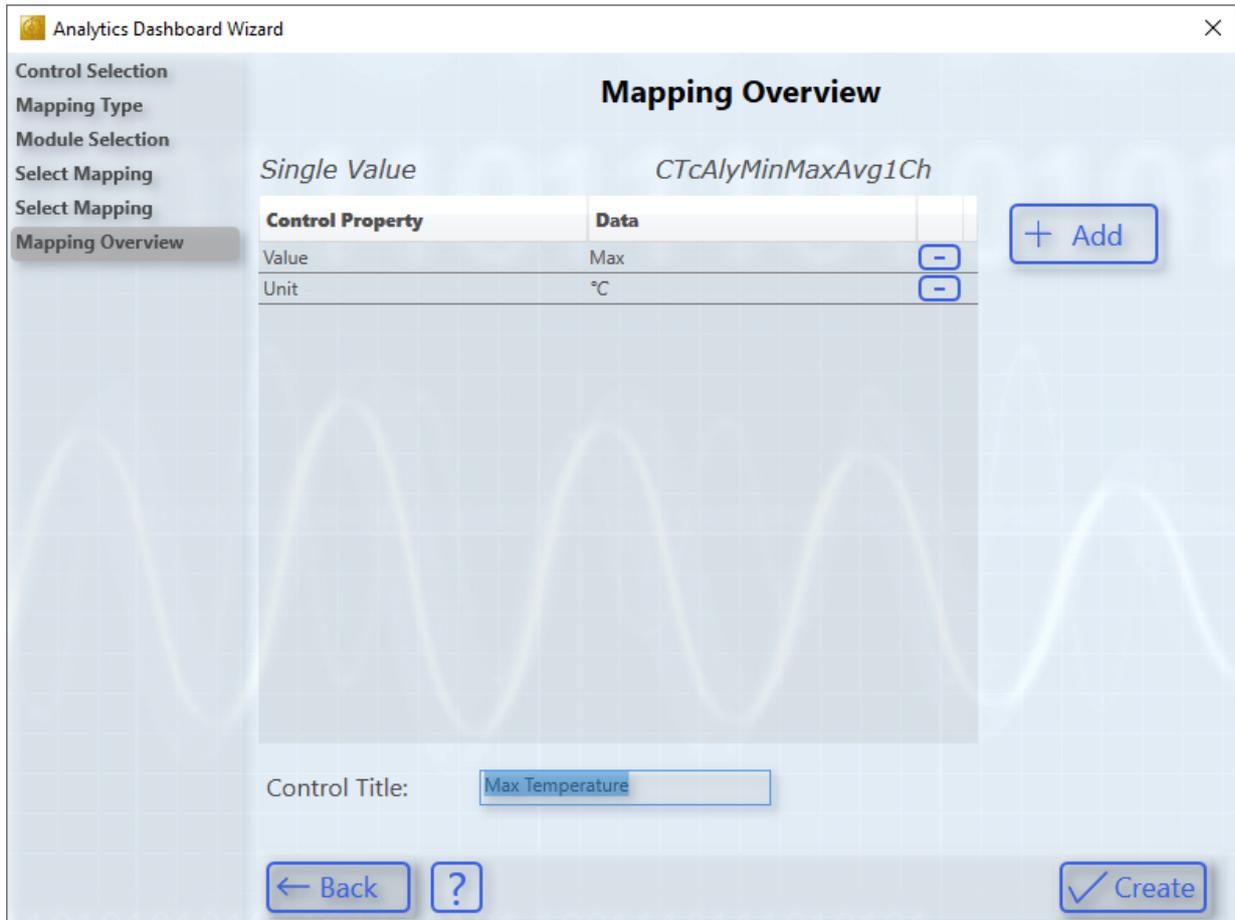
Control Title:

← Back ? ✓ Create

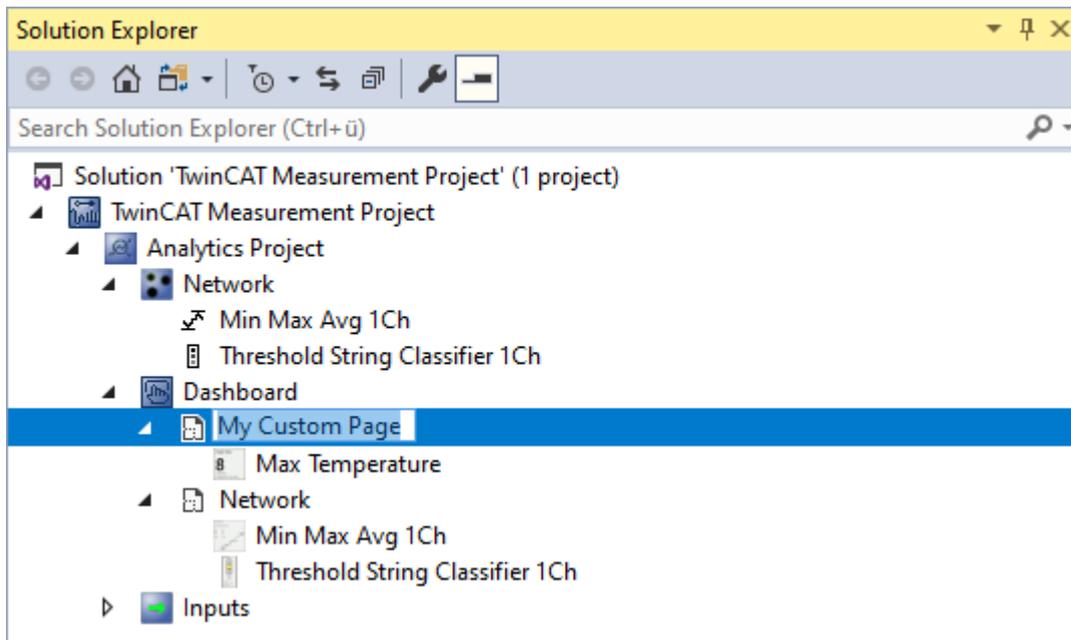
- Wählen Sie **Unit** aus und setzen Sie einen Haken bei **Set default value**, um einen statischen Wert zu vergeben. Auf der rechten Seite öffnet sich ein Textfeld, tragen Sie dort °C ein.



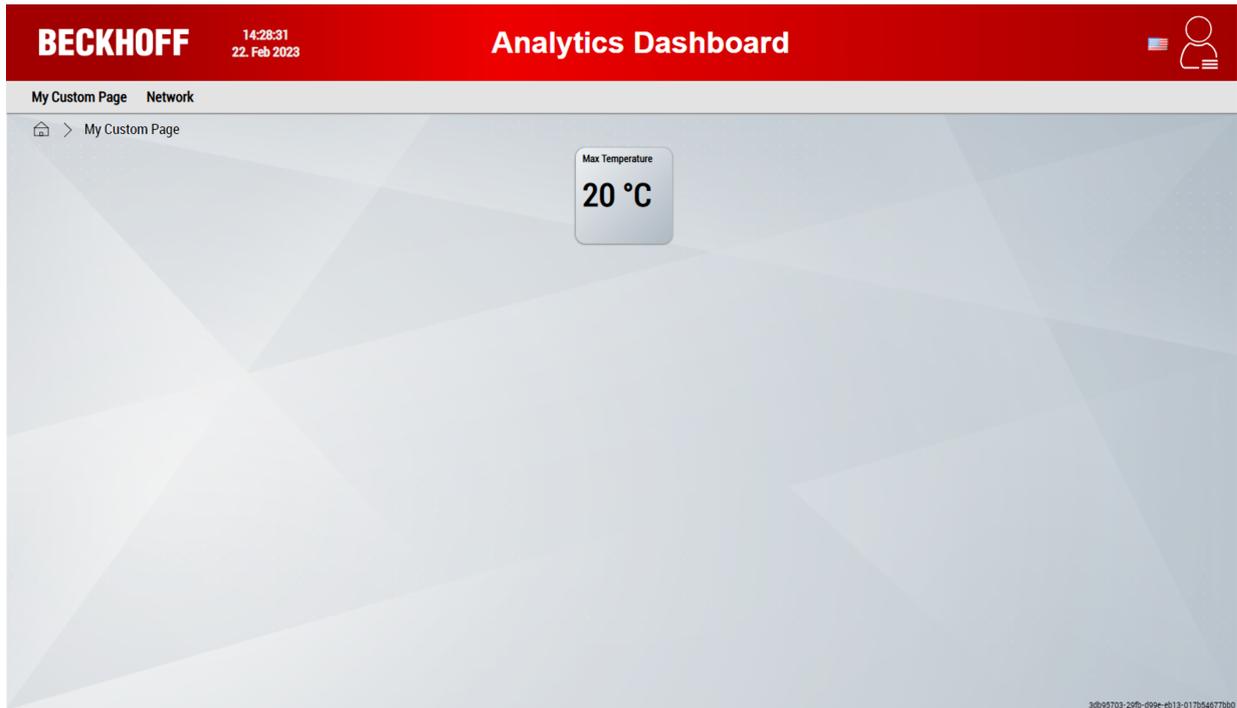
- Ändern Sie Titel des Controls in **Max Temperature** und fügen Sie über **Create** das Control-Mapping hinzu.



- Im Knoten **Dashboard** ist das erstellte Control **Max Temperature** sichtbar. Mit einem **Rechtsklick > Rename** können Sie den Content umbenennen.



⇒ Nach einer erfolgreichen HMI und SPS Generierung können Sie zuvor erstelle Seite **My Custom Page** über die Navigation öffnen. Dort sehen Sie das manuell angelegte Control.



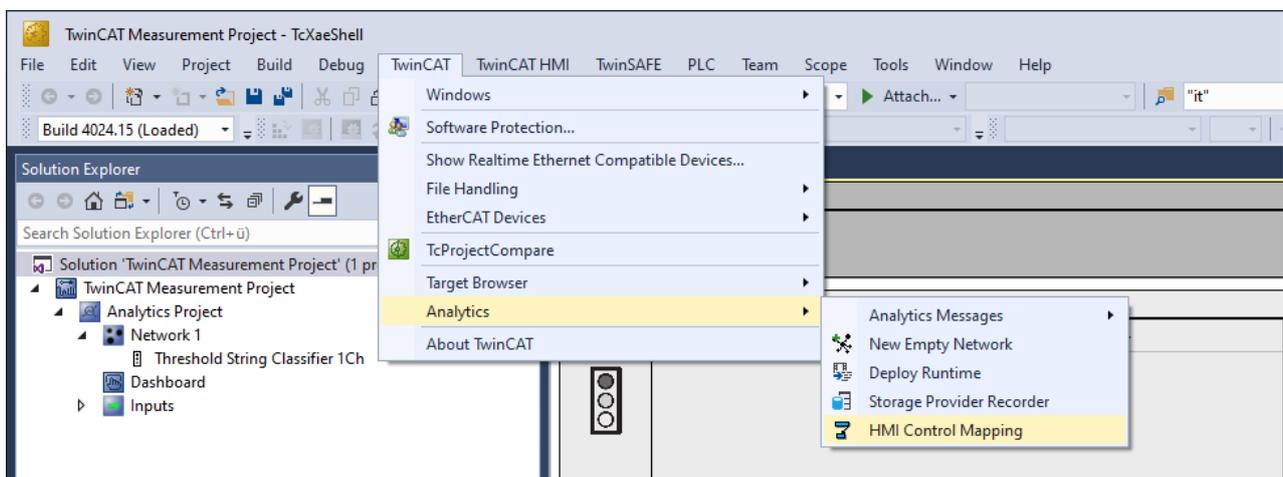
## 6.2.2 Angepasste und eigene Controls verwenden

### HMI Control Mapping Wizzard

Der HMI Control Mapping Wizard ermöglicht Folgendes:

1. Das Anfügen eigener Controls.
2. Das Mappen von Controls zu Modul-Klassen oder das Ändern vorhandener Mappings.
3. Das Mappen von Controls zu Modul-Instanzen oder Virtuellen Inputs (Analytics Projekt muss geöffnet sein)

Öffnen Sie den Wizard über den Reiter **TwinCAT > Analytics > HMI Control Mapping**.

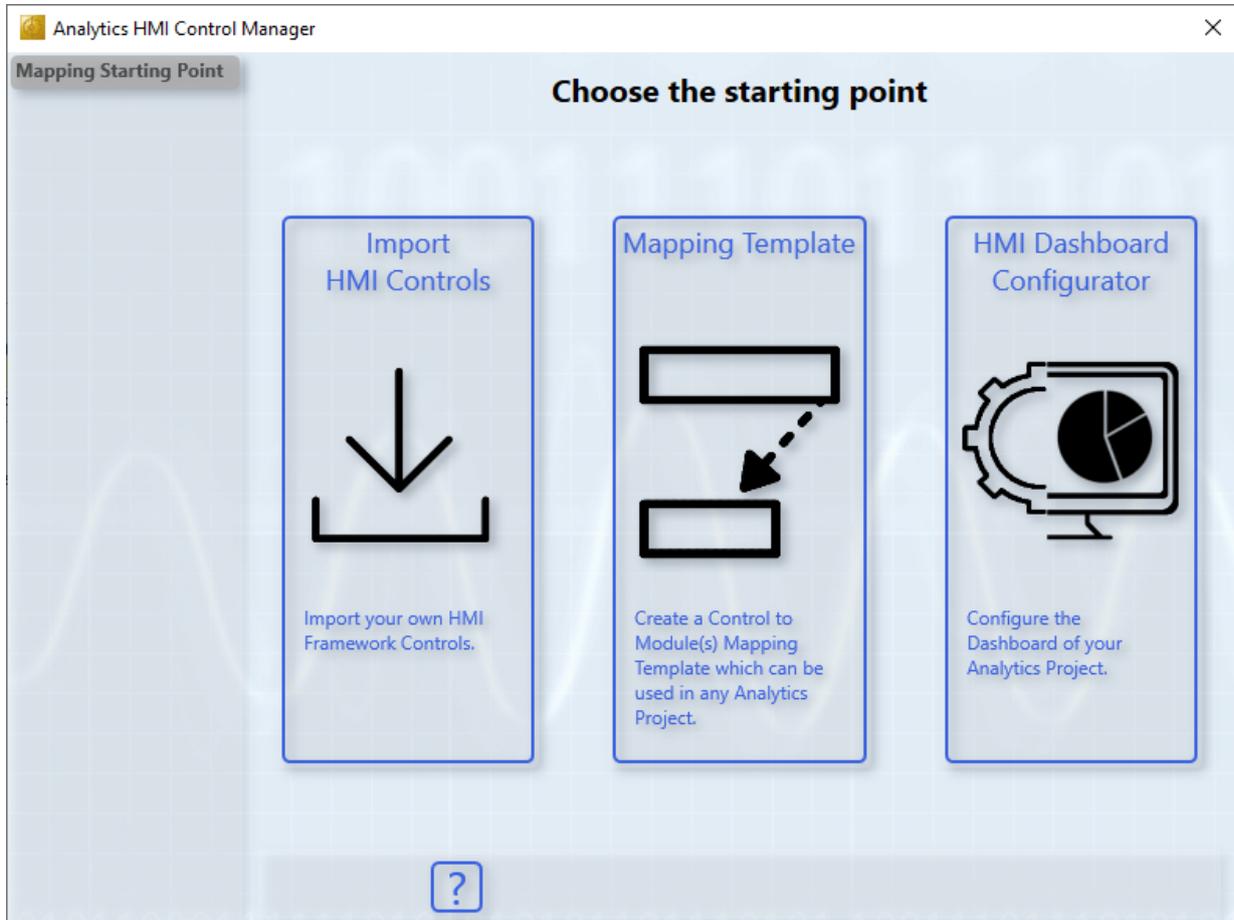


Über das Fragezeichen ? im Wizard wird Ihnen zusätzliche Hilfe angeboten.

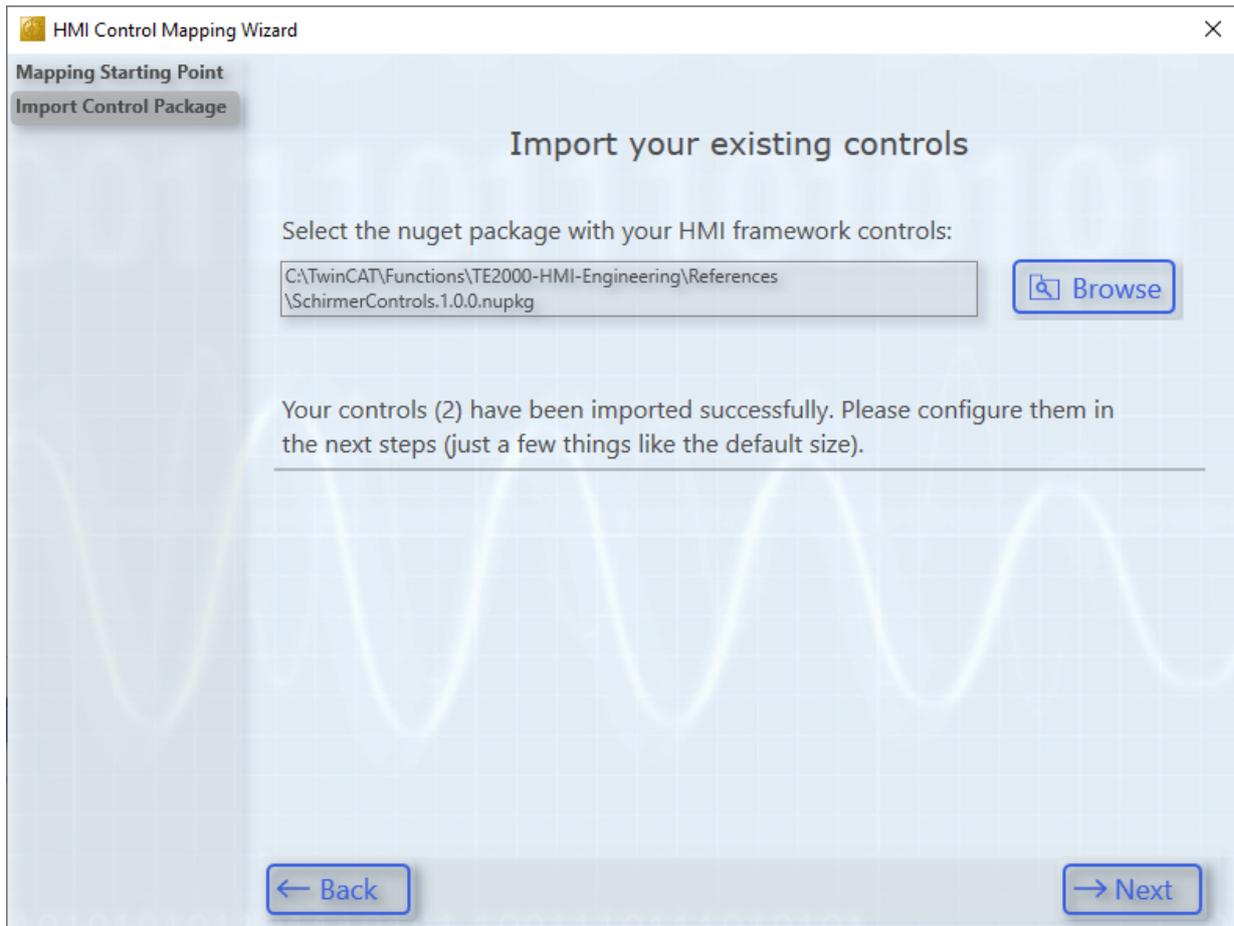
## 1. Anfügen eigener Controls

Die TwinCAT 3 HMI ermöglicht es, eigene HMI Framework Controls zu erstellen und diese als NuGet-Paket zu exportieren.

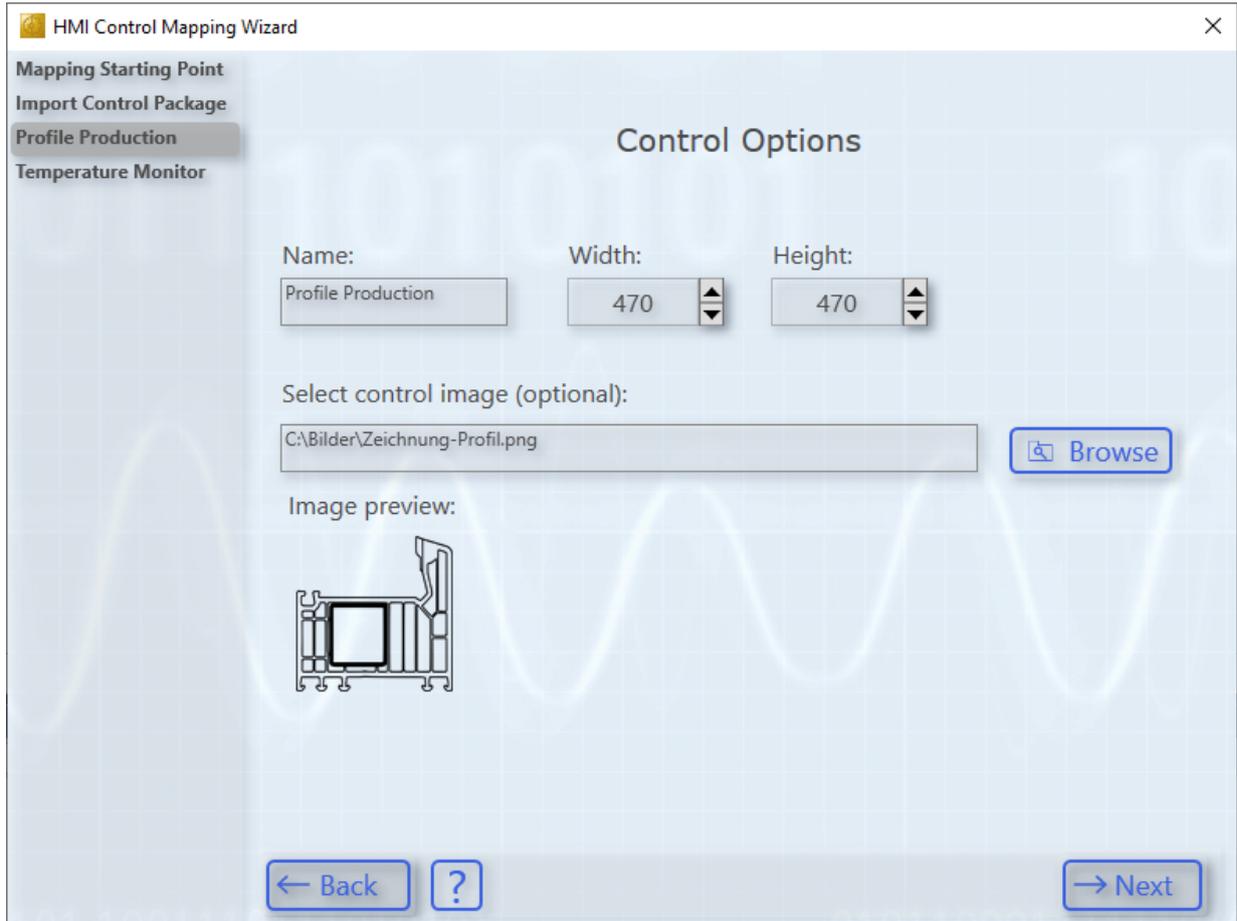
1. Um eigene Framework Controls den Analytics-Modulen zuzuordnen, klicken Sie auf **Import HMI Controls**



2. Wählen Sie das NuGet-Paket über **Browse** aus.



3. Wählen Sie für Ihre Controls als Nächstes einen Namen, eine Größe und ein Bild aus. Klicken Sie dann auf **Next** und bei dem letzten Control auf **Create**.



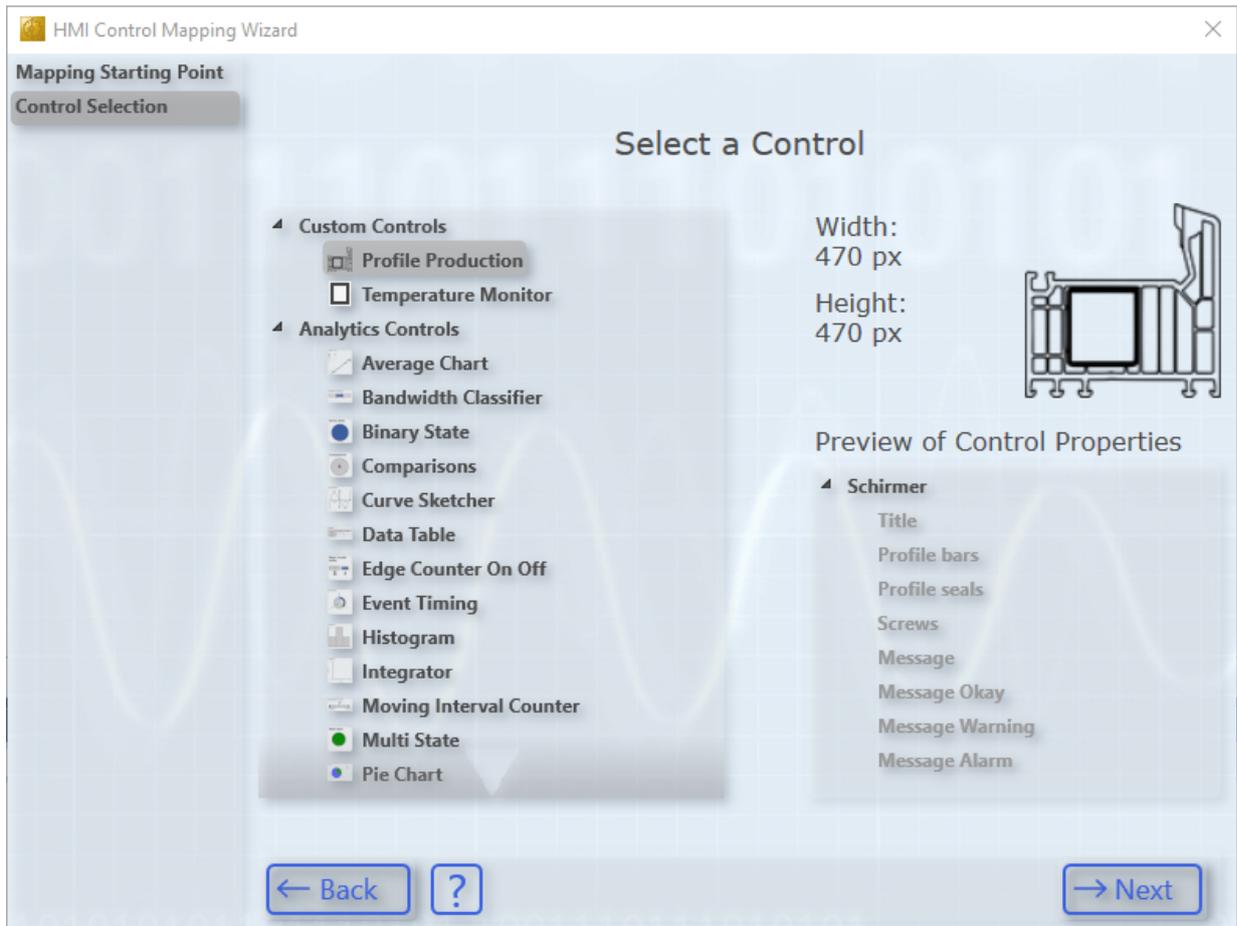
- ⇒ Sie werden automatisch weitergeleitet, sodass Sie ein Mapping zwischen Ihren Controls und Analytics Modulen erstellen können.

## 2. a) Mappen von Controls zu Modulen (weiterführend zu 1.)

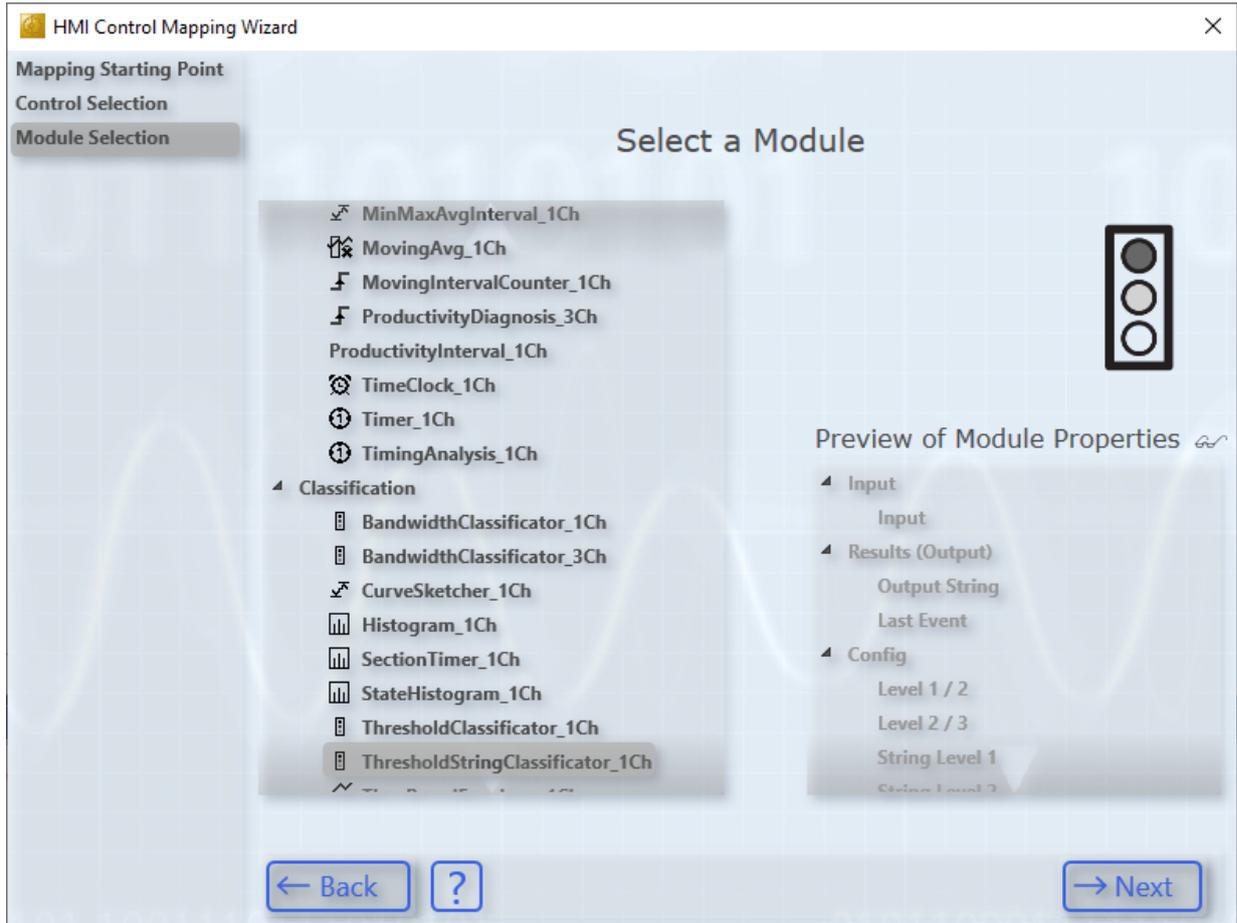


Unter 2. b) wird dieser Schritt nochmal anhand des „Binary State“ Control detaillierter erläutert.

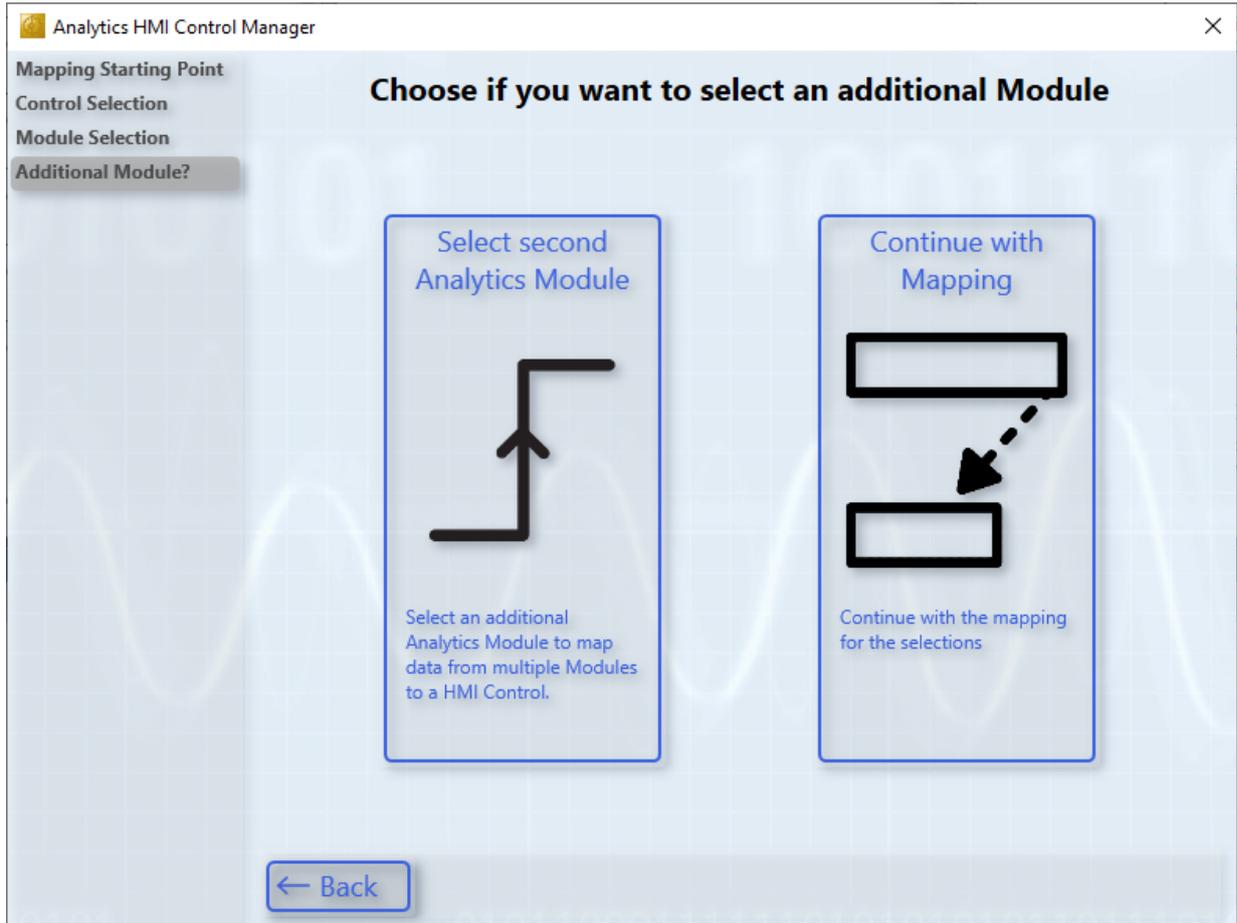
1. Wählen Sie nun Ihr Control aus.



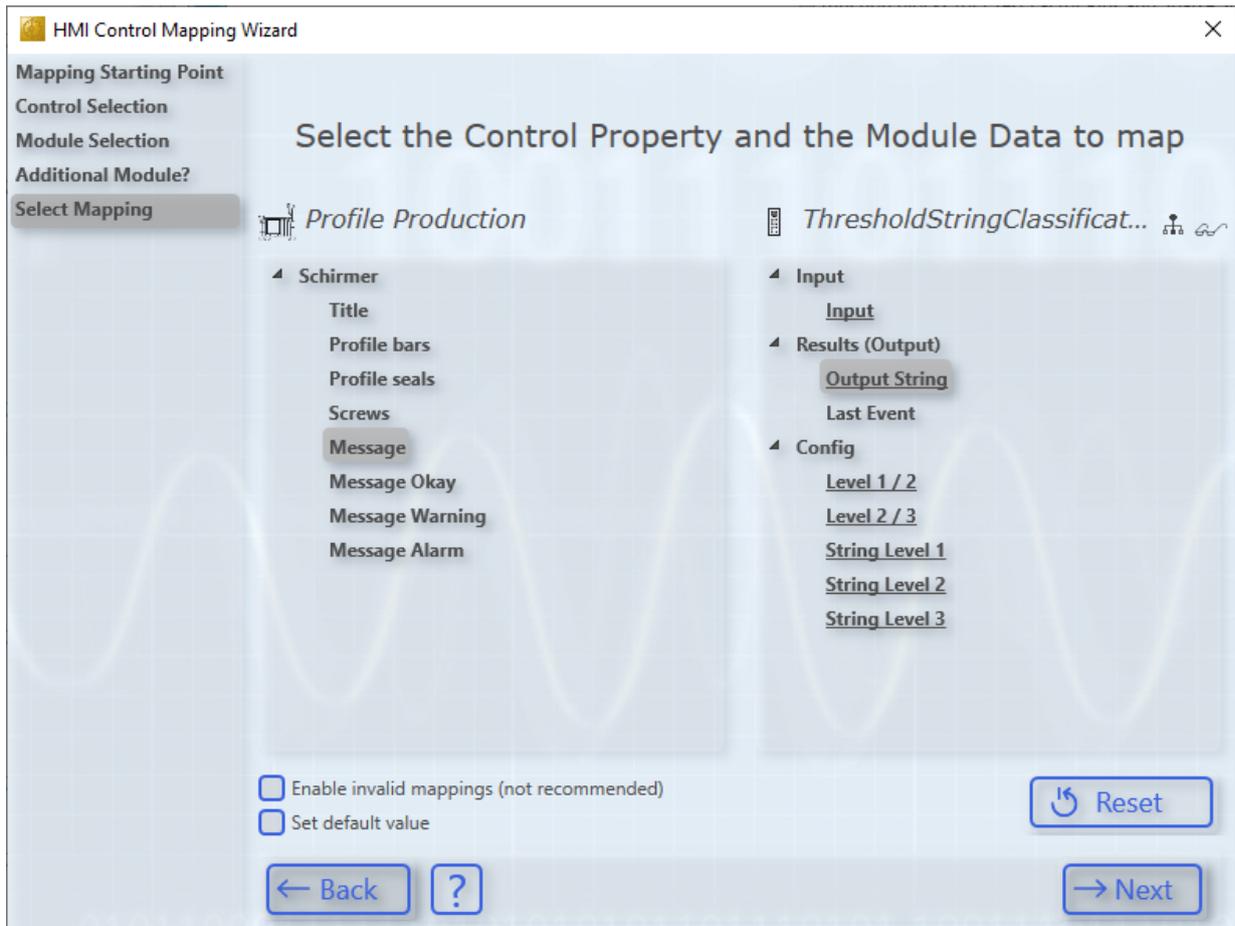
2. Wählen Sie nun das Modul aus, dem Sie das Control zuordnen wollen.



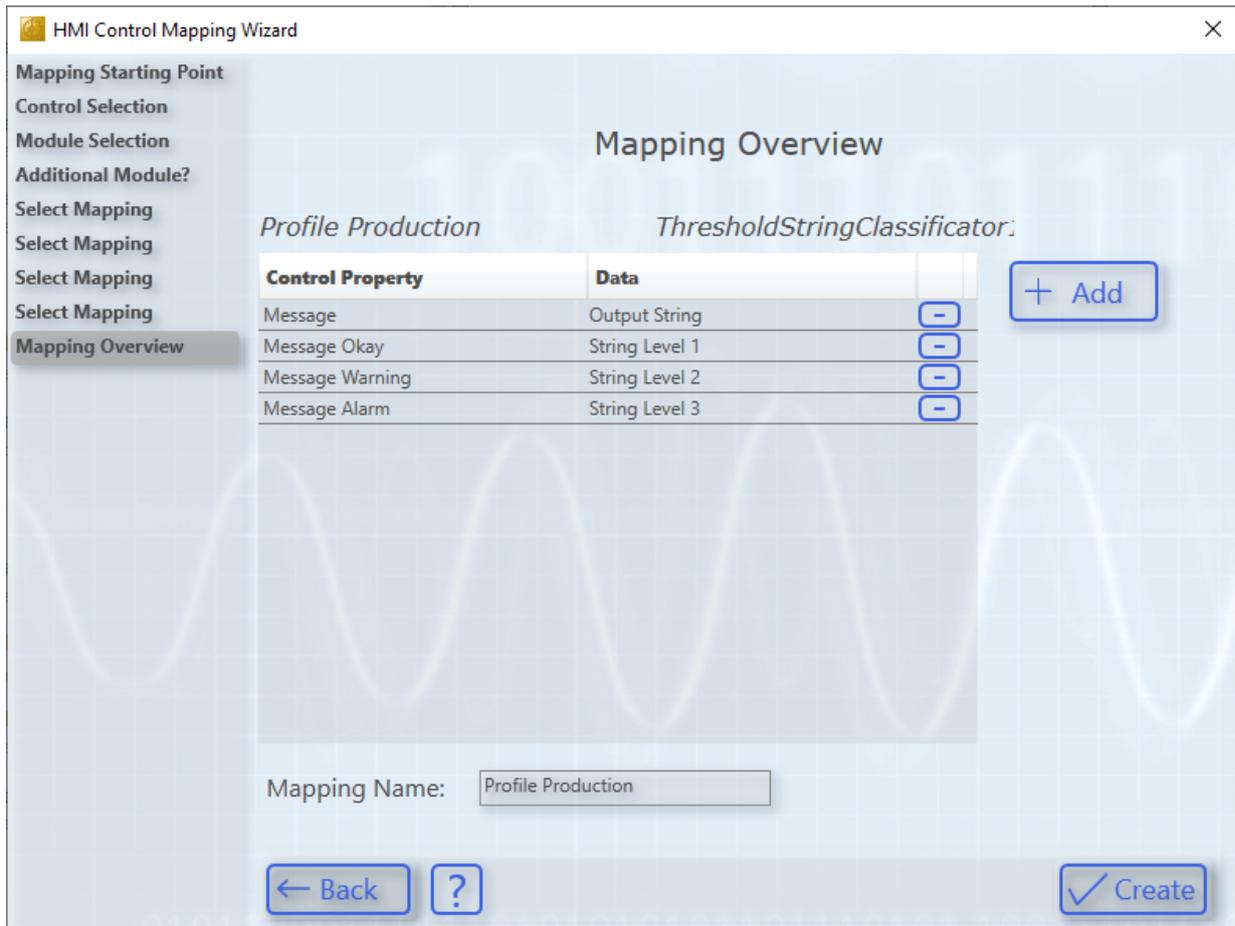
3. Wählen Sie **Continue with Mapping**.



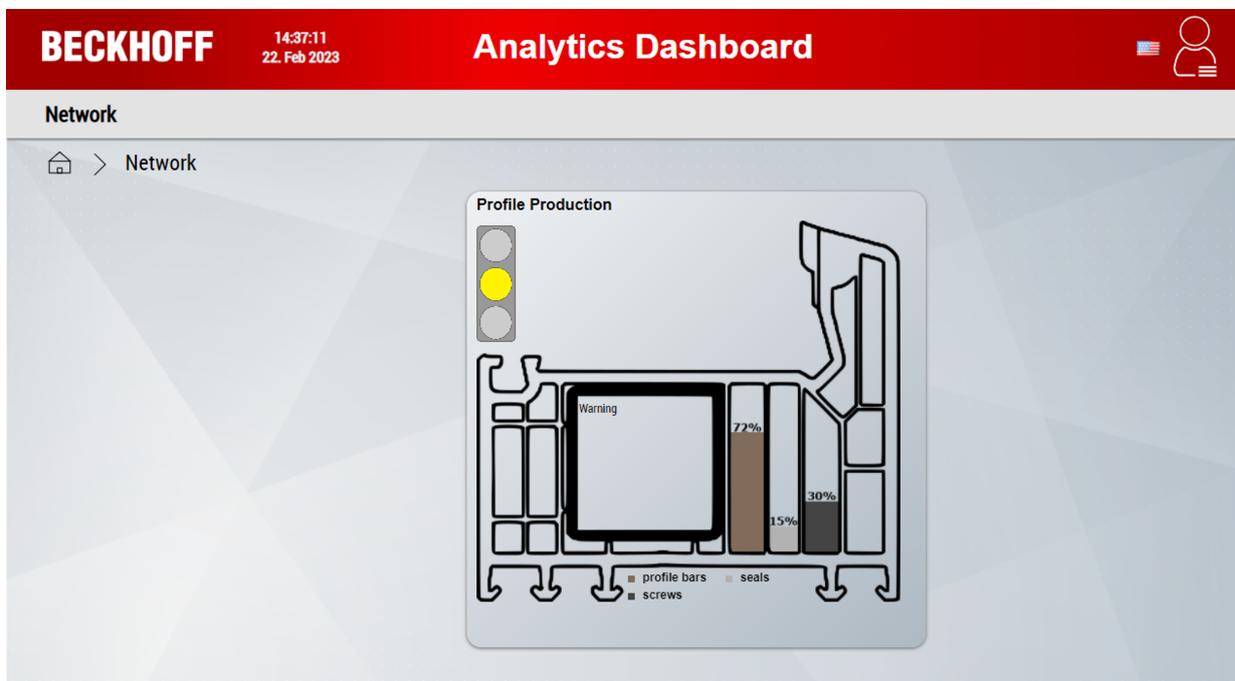
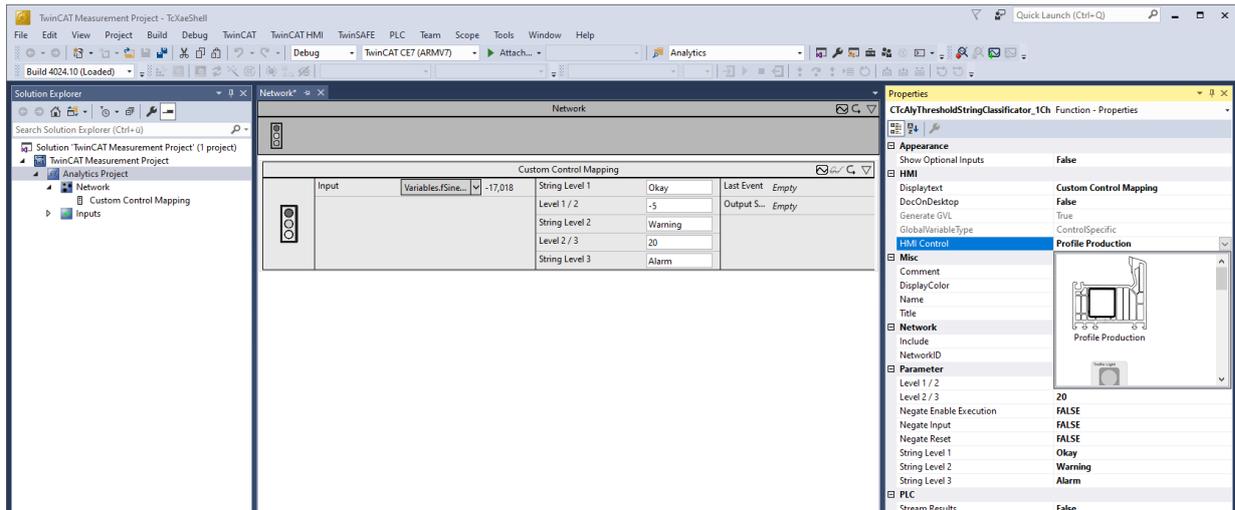
4. In den nächsten Schritten verbinden Sie die Control-Eingänge mit den Moduldaten.



5. Als Letztes klicken Sie auf **Create**, um das Mapping der Analytics Workbench hinzuzufügen.



6. Sie können nun den Wizard schließen und das Mapping wird automatisch für das Modul verfügbar sein. Wählen Sie es aus und generieren Sie sich Ihr Dashboard.

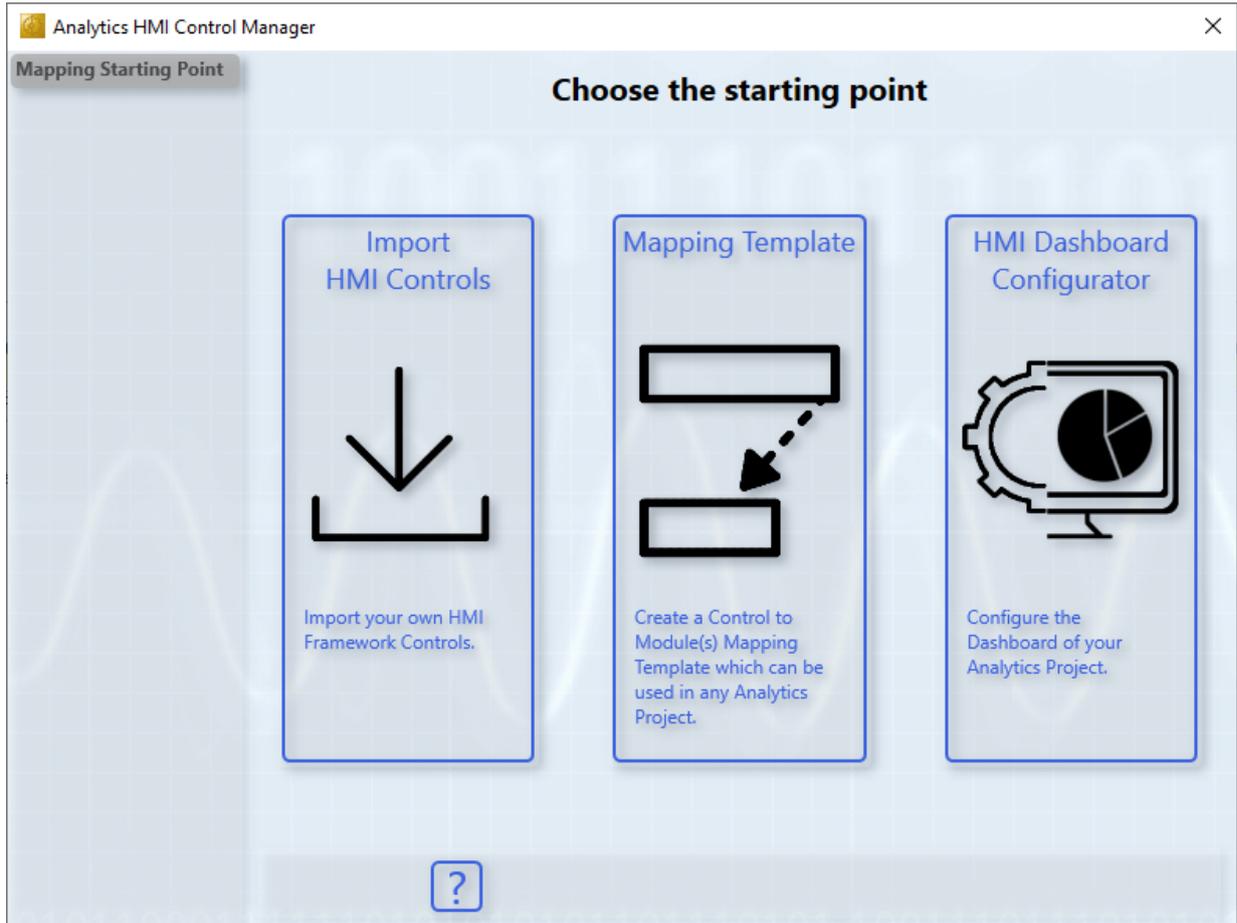


⇒ Das Dashboard ist fertig.

## 2.b) Mappen von Controls zu Modulen (ohne eigene Controls)

Analog zu 2.a) wird in diesem Abschnitt beschrieben, wie Sie ein Mapping zwischen einem Control und einem Modul durchführen. Als Beispiel wird ein bestehendes Analytics Control verwendet. Sie können dieses Beispiel also direkt ausprobieren.

1. Wählen Sie dafür den Punkt **Mapping Template**.

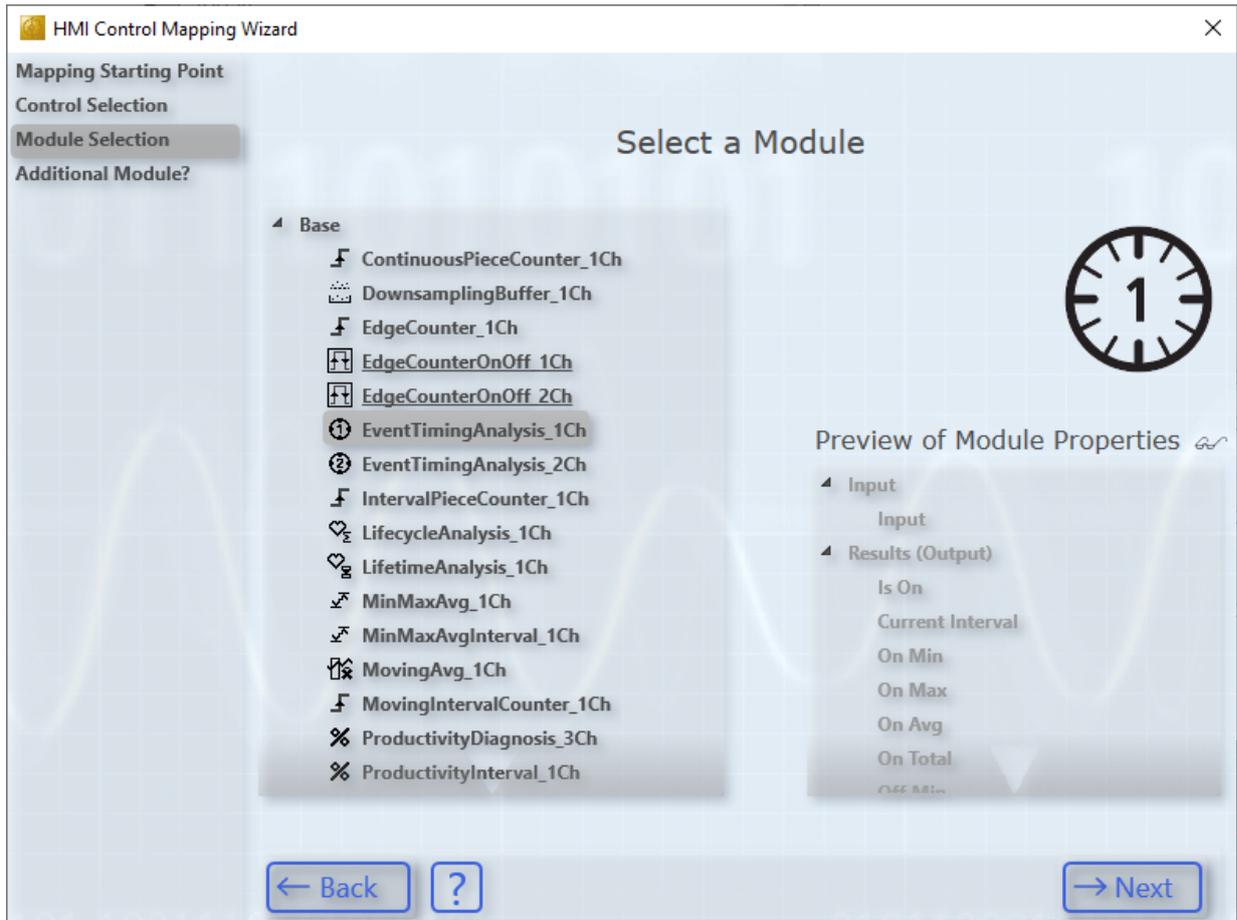


2. Wählen Sie ein Control aus. Auf der rechten Seite werden die Eigenschaften des Controls angezeigt. Ebenso wird angezeigt, wie groß das Control auf dem Dashboard sein wird.

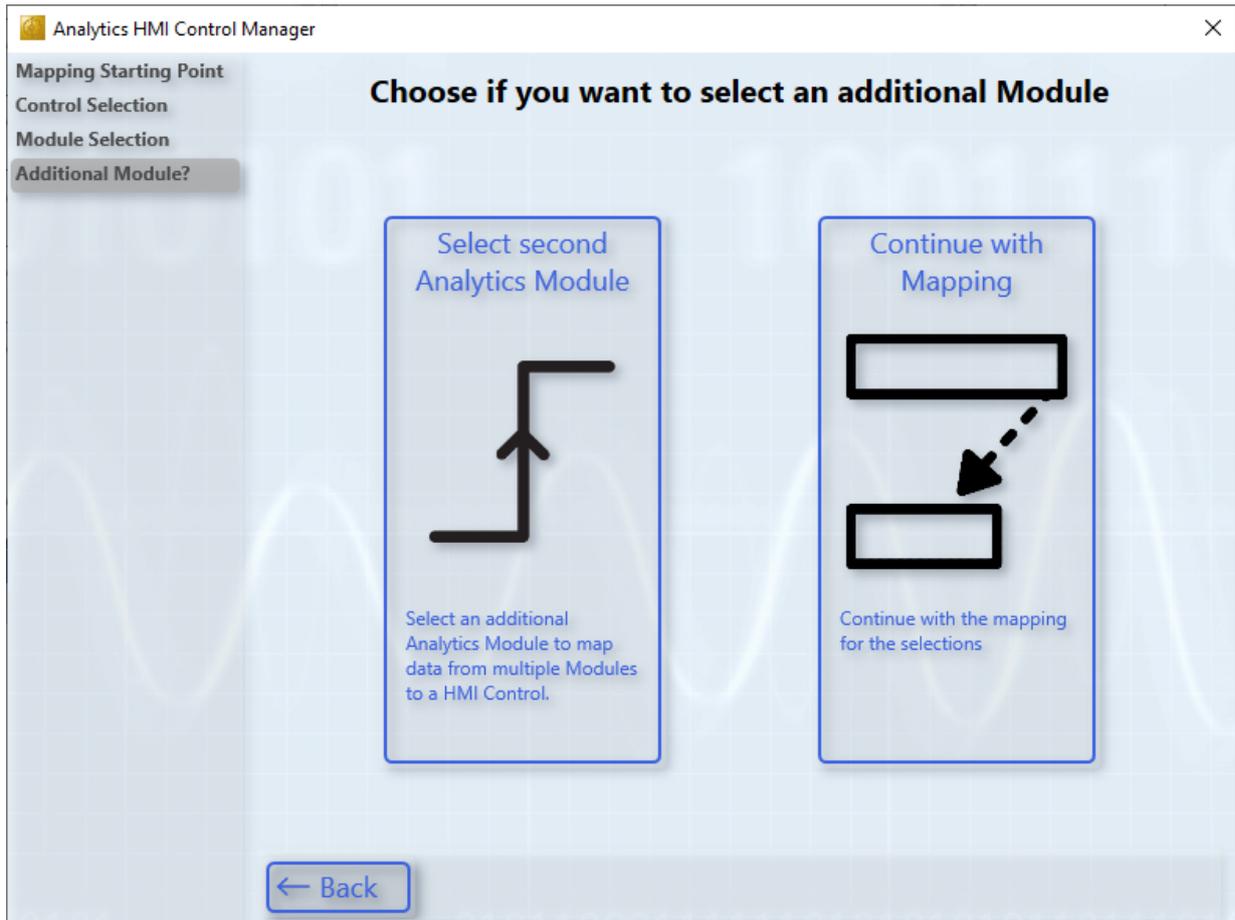


3. Wählen Sie das Modul aus, von dem Sie die Daten anzeigen lassen möchten. Es werden die Eingänge, Ausgänge und Parameter als Vorschau angezeigt, die anschließend mit den Eigenschaften des ausgewählten Controls verknüpft werden können.

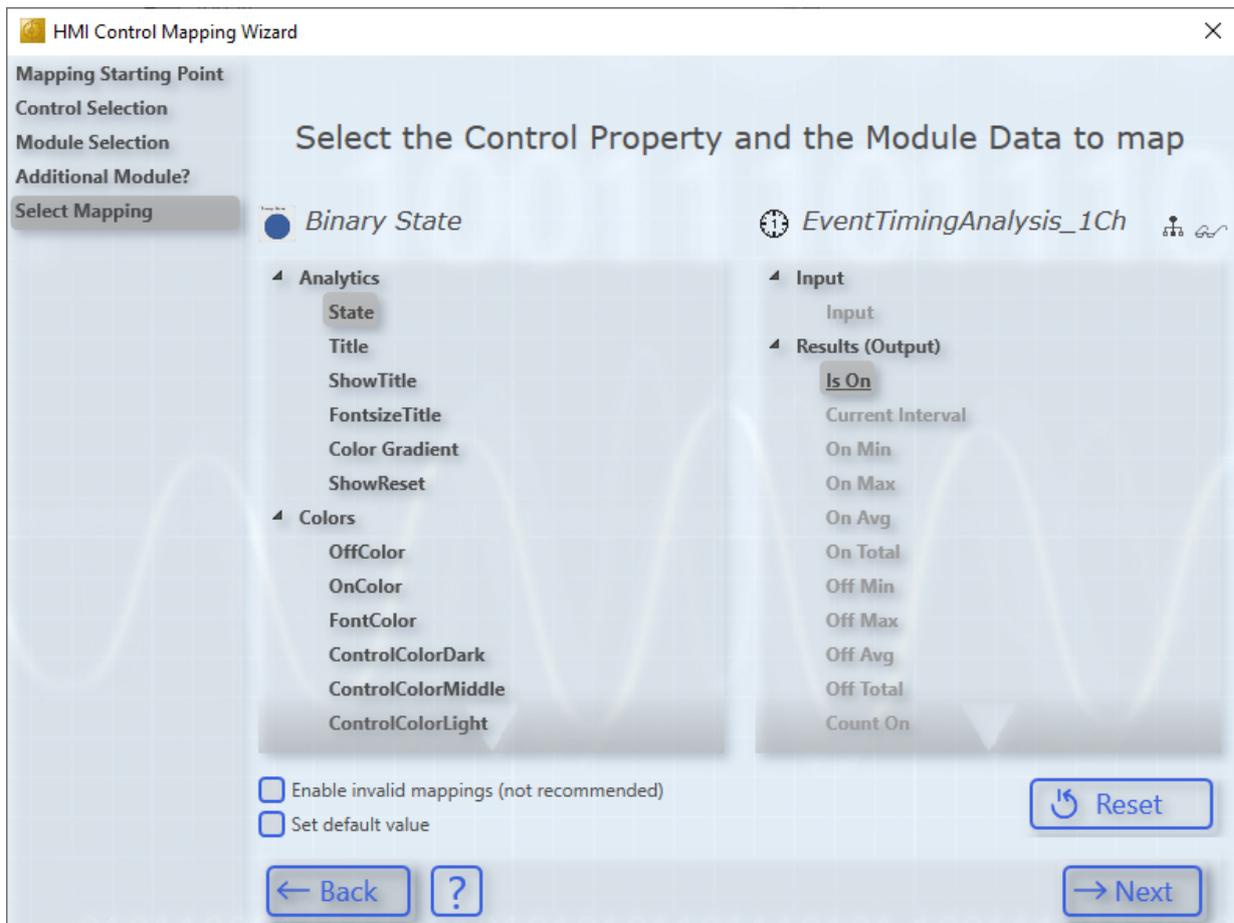
Bereits gemappte Module werden unterstrichen dargestellt. Diese Mappings können auch editiert werden.



4. Zukünftig kann ein Control auch von verschiedenen Modulen Daten darstellen. Da dies derzeit nicht möglich ist, wählen Sie **Continue with Mapping**.

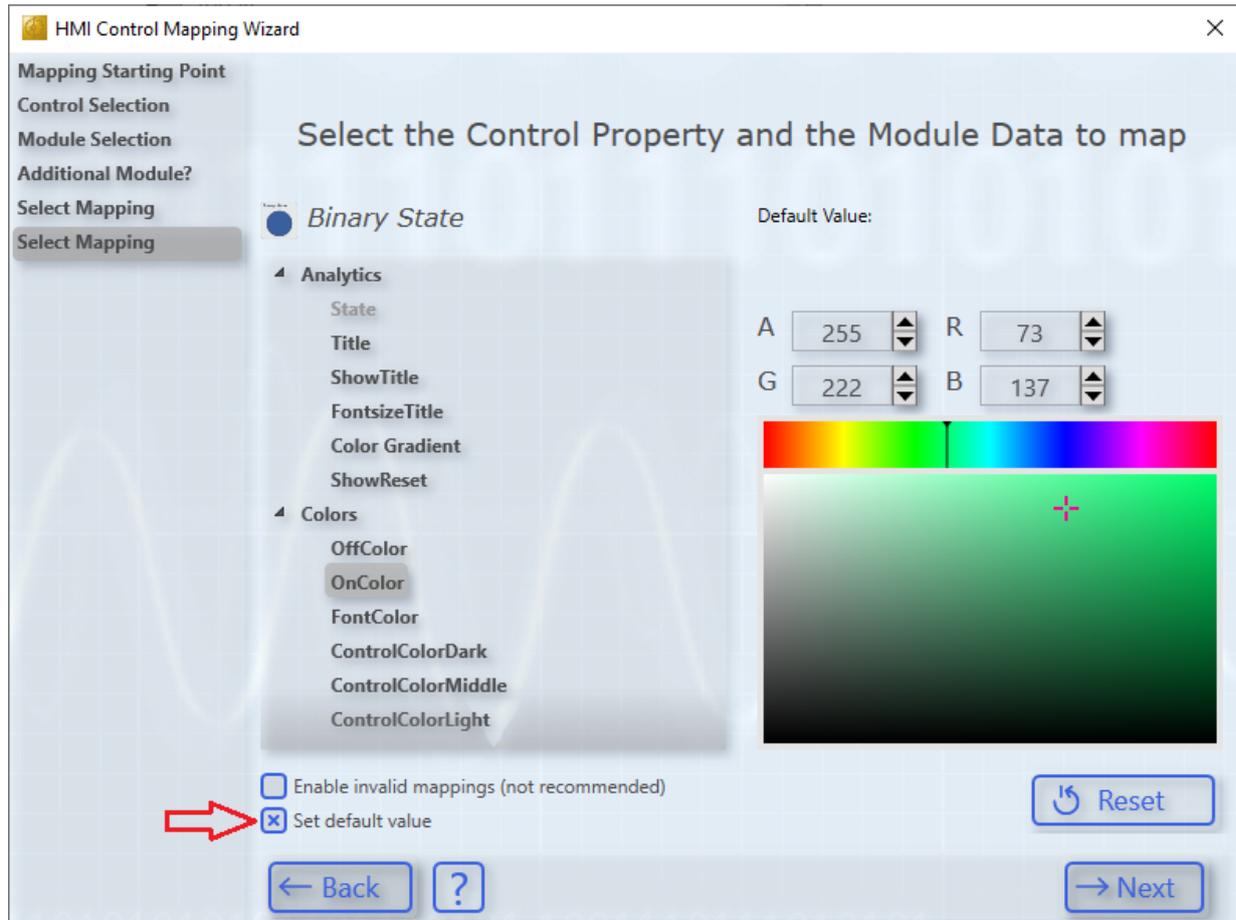


5. Wählen Sie als Nächstes eine Control Eigenschaft aus, die gemappt werden soll. Alle unmöglichen Variablen, die aufgrund ihres Datentyps nicht zugeordnet werden können, sind ausgegraut. Alle anderen können gemappt werden.  
Wählen Sie nun die Modul Variable aus, die Sie mit der Control Eigenschaft mappen wollen.



6. Ebenso können Sie die Checkbox **Set default value** aktivieren, um einen Standardwert dem Control zuzuweisen. Damit können beispielsweise Standardfarben geändert oder auch boolesche Werte wie „ShowTitle“ auf „False“ gesetzt werden, wenn in dem Analytics Control kein Titel angezeigt werden soll.

In diesem Beispiel wird die „OnColor“ geändert. Anhand des Datentyps wird eine spezifische Auswahlmöglichkeit zur Verfügung gestellt.



7. Soweit Sie keine weiteren Einträge mehr über **Add** hinzufügen wollen, können Sie über **Create** das Mapping vollenden. Es wird empfohlen einen aussagekräftigen Mapping- Namen zu vergeben.

**HMI Control Mapping Wizard** [Close]

Mapping Starting Point  
 Control Selection  
 Module Selection  
 Additional Module?  
 Select Mapping  
 Select Mapping  
**Mapping Overview**

### Mapping Overview

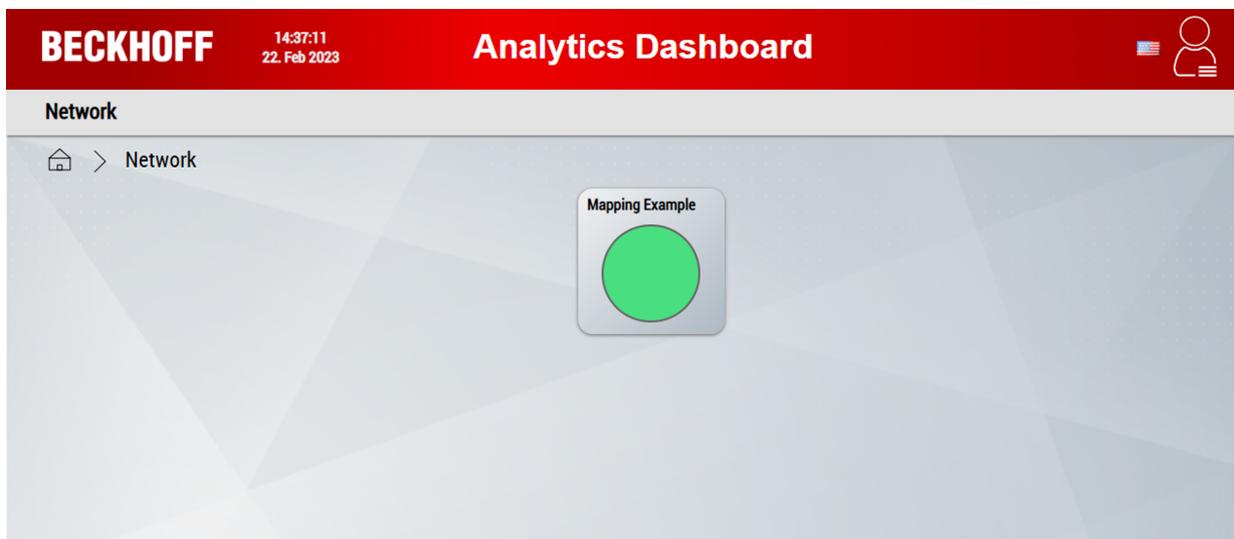
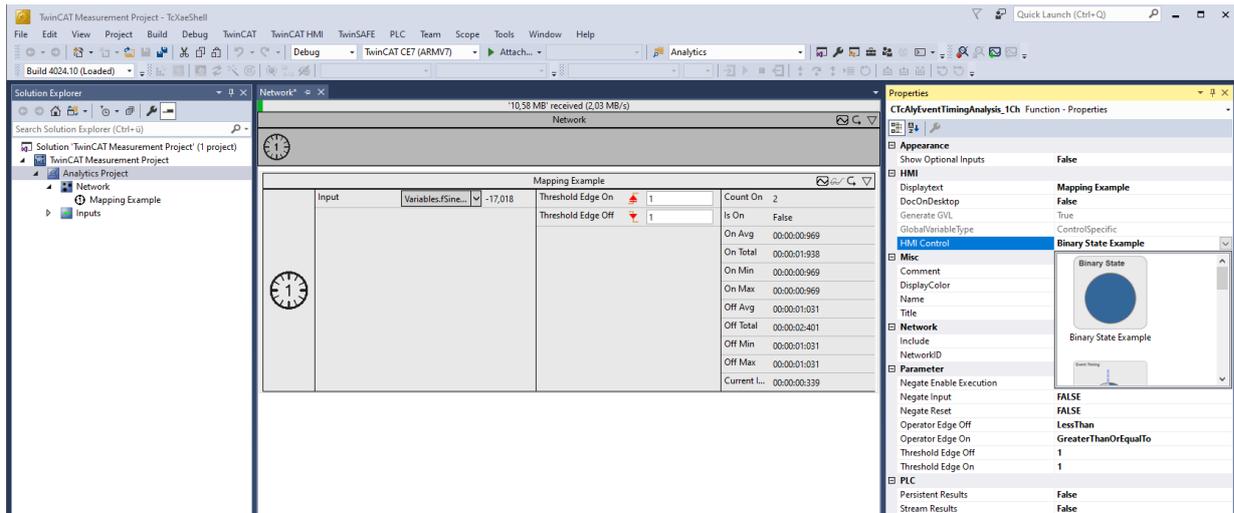
Control Property	Data	
State	Is On	-
OnColor	color: "rgba(73, 222, 137, 1)"	-

+ Add

Mapping Name:

← Back   ?   ✓ Create

8. Sie können den Wizard schließen und das Mapping wird automatisch für das Modul verfügbar sein. Wählen Sie es aus und generieren Sie sich Ihr Dashboard.

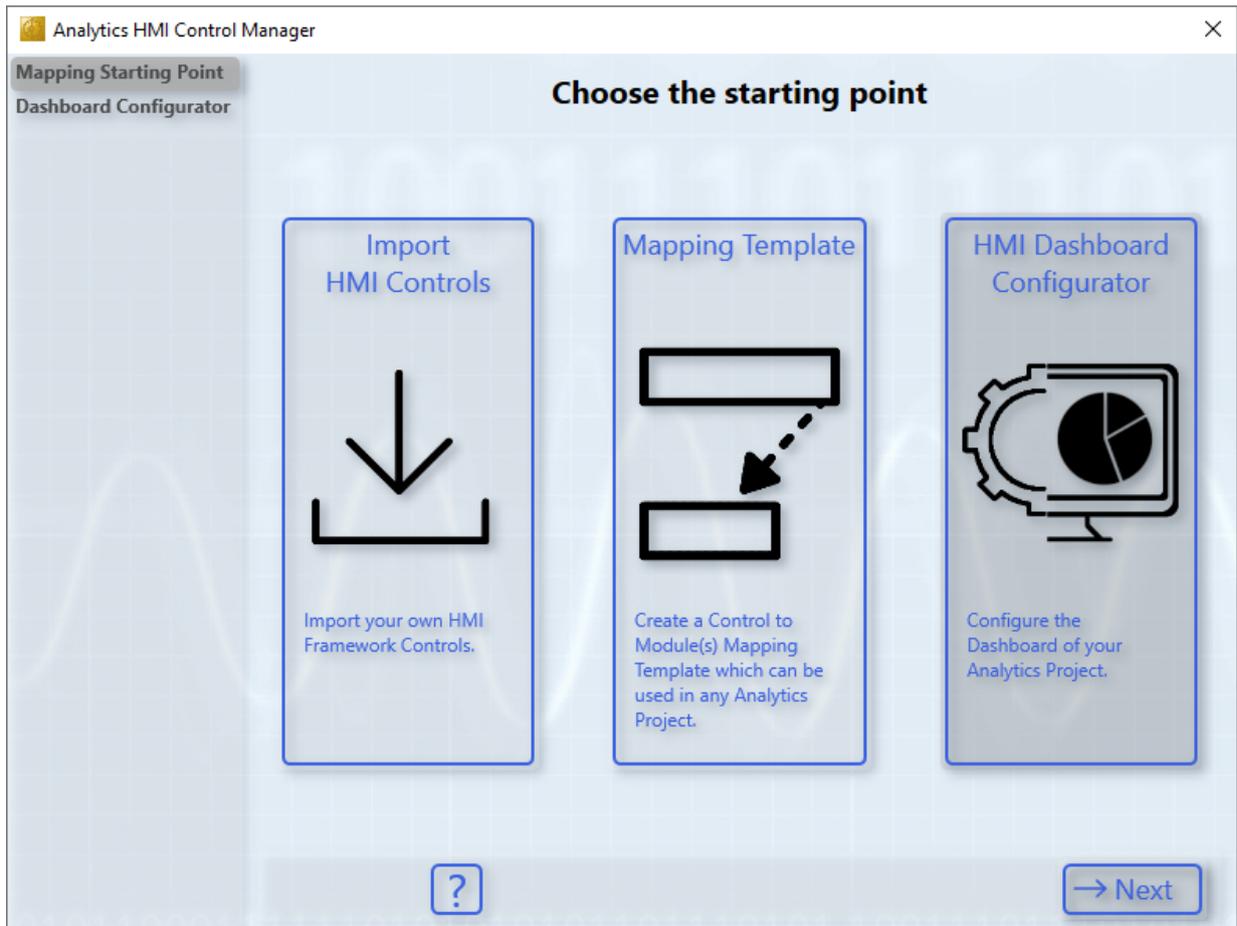


⇒ Das Dashboard ist fertig.

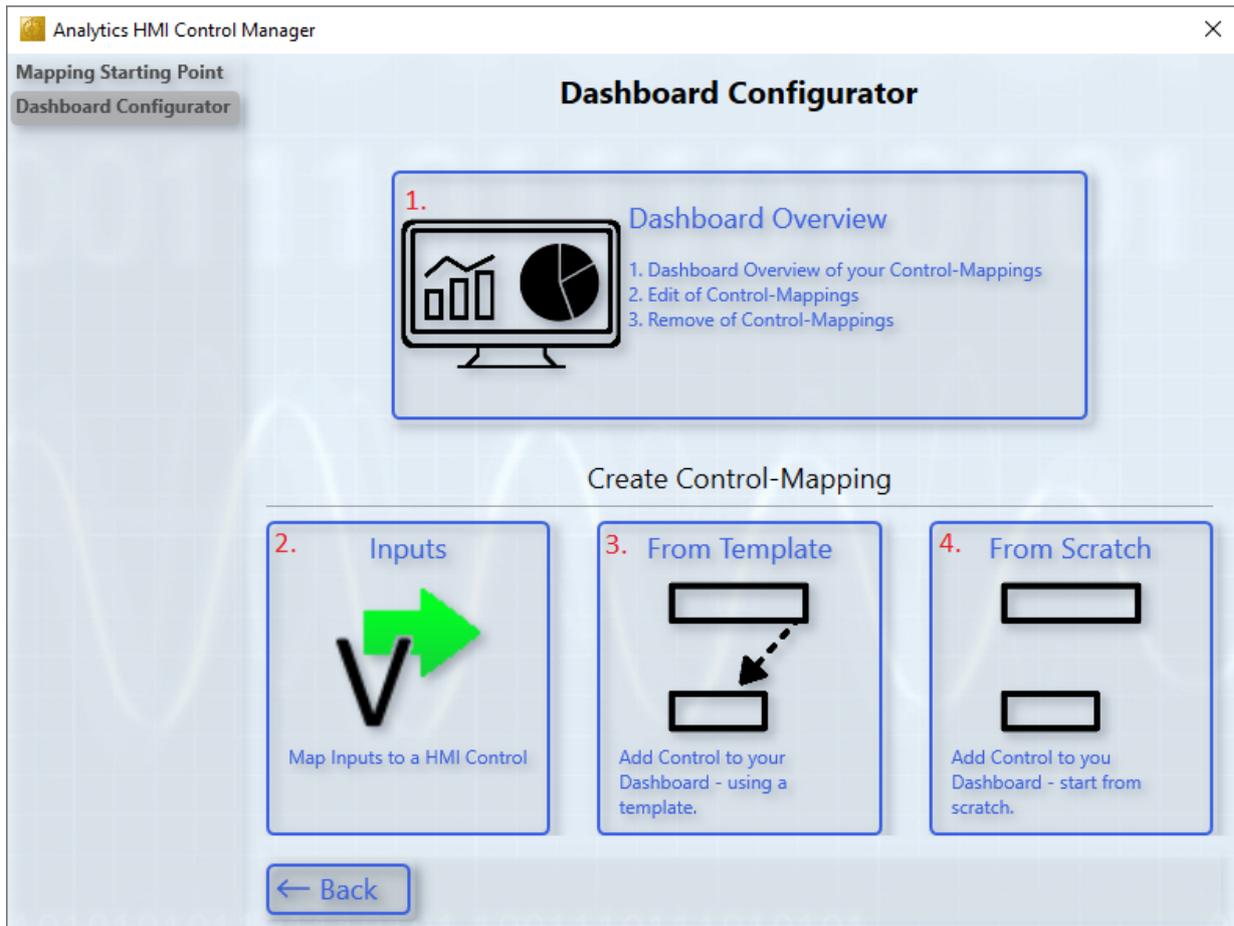
### 3. Mappen von Controls zu Modul-Instanzen

Neben dem Mappen von Controls zu Modul-Klassen können auch direkt die Modul-Instanzen eines Projekts mit Controls verknüpft werden. Dies ist sowohl über den Dashboard Aufbau und Inhalte im Analytics Projekt verwalten [[▶ 113](#)] als auch über den Wizard möglich.

1. Klicken Sie zu Beginn des Wizards auf **HMI Dashboard Configurator**.



2. Dort haben Sie mehrere Auswahlmöglichkeiten.



1. Übersicht und Bearbeitungsmöglichkeiten über alle Controls
  - Diese Funktion ist ebenfalls über das Doppelklicken eines Controls im Solution Explorer möglich.
2. Ein Control hinzufügen, dass Ihre Eingangsdaten anzeigt
  - Diese Funktion ist ebenfalls über den [Dashboard Aufbau und Inhalte im Analytics Projekt verwalten](#) [113] möglich.
3. Ein Control hinzufügen, dass Daten von Modulen anzeigt (Auswahl über ein Template)
  - Diese Funktion ist für ein einzelnes Modul auch über das *Properties* Fenster einstellbar. Zur Darstellung von Daten aus mehreren Modulen ist dies der einzige Weg.
  - Sie sehen direkt welche Daten aus dem Template verknüpft sind und können diese direkt anpassen.
4. Ein Control hinzufügen, dass Daten von Modulen anzeigt (komplett Manuell)
  - Diese Funktion ist ebenfalls über den [Dashboard Aufbau und Inhalte im Analytics Projekt verwalten](#) [113] möglich.

Für die einzelnen Punkte müssen Sie den Schritten des Wizards folgen. Die Schritte sind gleich bzw. sehr ähnlich zu Abschnitt „2. Mappen von Controls zu Modulen“ aufgebaut.

### 6.2.3 Benutzerverwaltung und Zugriffsrechte konfigurieren

Über das Benutzermanagement lassen sich individuell Benutzer und Benutzergruppen generieren. Für Benutzergruppen können Zugriffsrechte auf Contents und Controls definiert werden.

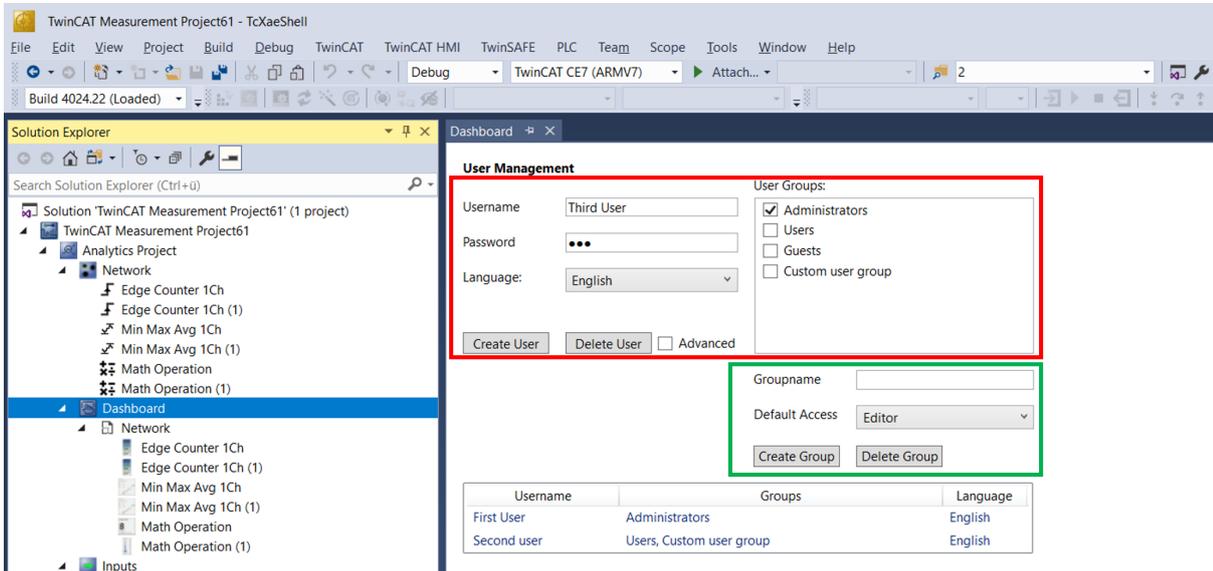


Beachten Sie, dass das Benutzermanagement und die Zugriffsrechte erst ab der Dashboard-Version 2.0 in das HMI-Projekt übernommen werden.

### 6.2.3.1 Definition im Analytics Projekt

Definierte Benutzer, Benutzergruppen und Zugriffsrechte werden bei der Code-Generierung automatisch in das HMI-Projekt übernommen. Im Webbrowser lassen sich nach der Generierung Änderungen an den Benutzern und Zugriffsrechten vornehmen.

#### 6.2.3.1.1 Benutzerverwaltung



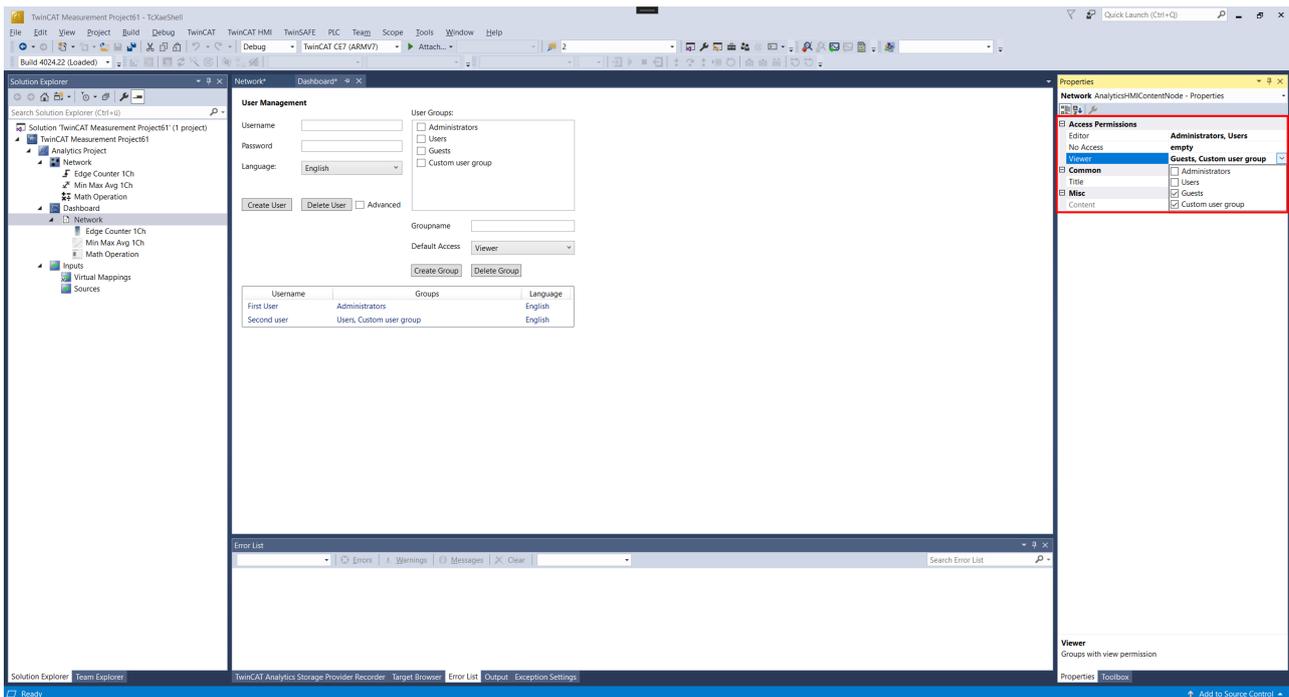
Unter dem Dashboard-Knoten des Analytics-Projekts können Benutzer (rot umrandet) angelegt werden. Für einen neuen Benutzer müssen ein Benutzername und Passwort sowie eine Sprache festgelegt werden. Zusätzlich gilt es dem Benutzer eine oder mehrere Benutzergruppen zuzuweisen, wodurch der Benutzer die Zugriffsrechte der jeweiligen Gruppe(n) erhält.

Standardmäßig gibt es folgende Benutzergruppen:

- Administratoren
- Users
- Guests

Für eine genauere Einteilung lassen sich weitere Benutzergruppen erstellen (grün umrandet). Hierfür muss ein Name und die Zugriffsberechtigung eingegeben werden.

### 6.2.3.1.2 Zugriffsrechte konfigurieren



Für Objekte unterhalb des Dashboard-Knotens (Contents und Controls) können Zugriffsrechte konfiguriert werden. Über die Access-Permissions (rot umrandet) kann für jede Benutzergruppe das Zugriffsrecht festgelegt werden.

Folgende Zugriffsrechte stehen zur Verfügung:

- Editor: Zugriff auf das Objekt gestattet, zudem können Änderungen (wie Layout-Anpassungen bei Content-Objekten) vorgenommen werden.
- Viewer: Zugriff auf das Objekt gestattet – jedoch können keine Änderungen vorgenommen werden.
- No Access: Kein Zugriff auf das Objekt und die dem Objekt im Solution-Explorer untergeordneten Objekte gestattet.



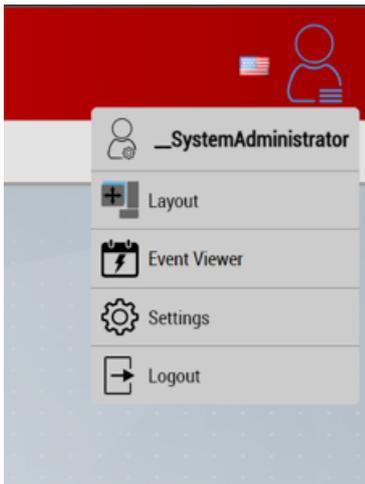
Zugriffsrechte für Content-Objekte lassen sich nachträglich über den Webbrowser anpassen.

### 6.2.3.2 Anpassen der Benutzer und Zugriffsrechte im Webbrowser

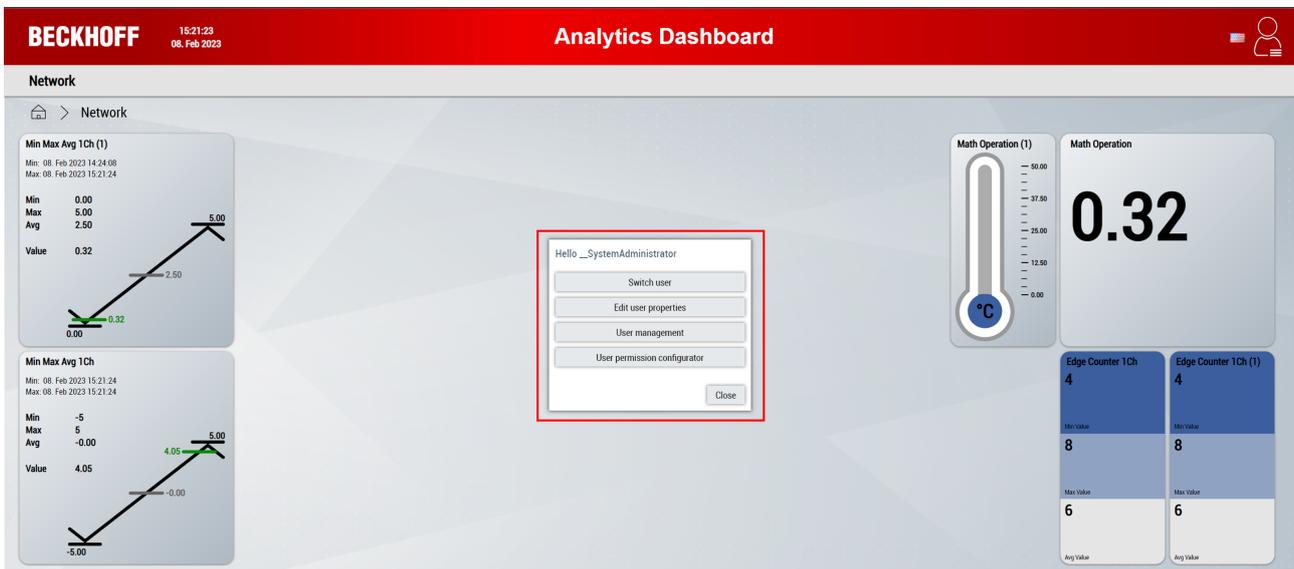


Beachten Sie, dass das Menü-Control sowie die Konfiguration von Benutzern und Zugriffsrechten erst ab der Dashboard-Version 2.0 verfügbar sind.

## Benutzer konfigurieren



Über das Menü-Control können Änderungen an den Benutzern und den Zugriffsrechten vorgenommen werden. Klicken Sie hierfür bitte auf Ihren Benutzernamen (ganz oben im Menü).



Über das Popup (rot umrandet) stehen folgende Optionen zur Verfügung:

- **Switch user:** Wechsel des angemeldeten Benutzers
- **Edit user properties:** Ändern der Eigenschaften des eigenen Benutzers
- **User management:** Erzeugen von neuen Benutzern, entfernen existierender Benutzer und festlegen von Benutzereigenschaften
- **User permission configurator:** Konfigurieren der Zugriffsrechte von Benutzergruppen auf Content-Objekte

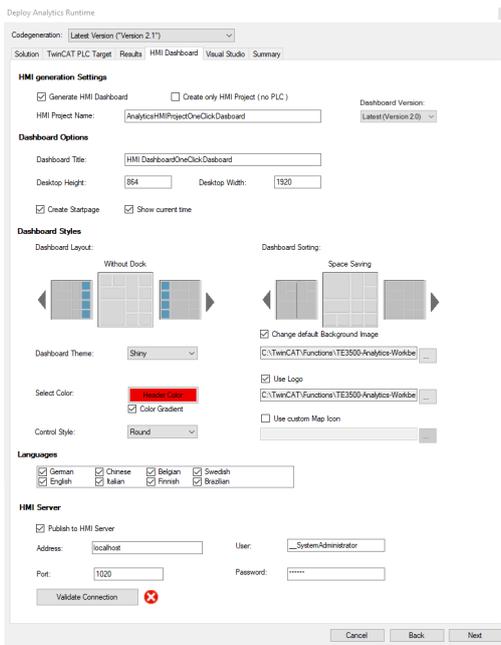


Beachten Sie, dass beim Erzeugen neuer Benutzer individuell konfigurierte Layouts des erzeugenden Benutzers übernommen werden.

## 6.2.4 Dashboard-Konfiguration

### Karteireiter HMI Dashboard

Der Karteireiter **HMI Dashboard** enthält alle Konfigurationen für das Dashboard.



## Konfiguration

### HMI generation Settings

Generate HMI Dashboard	Aktiviert die Generierung des HMI Dashboards, falls aktiviert. Damit werden die Checkboxes <b>Create Bootproject</b> und <b>Activate PLC Runtime</b> in dem Karteireiter <b>TwinCAT PLC Target</b> automatisch aktiviert/deaktiviert.
Create only HMI Project (No PLC)	Erstellt nur ein HMI-Projekt und kein SPS-Projekt, falls aktiviert.
HMI Project Name	Name des TwinCAT HMI-Projekts.
Dashboard Version	Auswahl der Dashboard-Version.

### Dashboard Options

Dashboard Title	Titel des HMI Dashboards, der im Header des Dashboards angezeigt wird.
Desktop Height	Höhe des Zielbildschirms in Pixeln.
Desktop Width	Breite des Zielbildschirms in Pixeln.
Create Startpage	Erstellt eine Startseite für das Dashboard, auf der eine Karte mit allen Maschinenstandorten angezeigt wird. Die Standortdaten werden aus den Maschinenverwaltungsdaten übernommen.
Show current time	Erstellt eine Uhr im Dashboard-Header, die die aktuelle Ortszeit anzeigt.

### Dashboard Styles

Dashboard Layout	Legt das Layout des Dashboards fest. Dock erfordert, dass die „Dock“-Eigenschaft eines Moduls TRUE ist. Dock Left: Fixierte linke Spalte Dock Right: Fixierte rechte Spalte Without Dock: Keine fixierte Spalte
Dashboard Sorting	Legt die Sortierung des Dashboards fest. Space Saving: Ordnet die Controls ohne Freiraum an. Control Type: Ordnet die Controls nach Control-Typ an. Control Size: Ordnet die Controls nach Größe von groß nach klein an. Filled: Ordnet die Controls so an, dass der gesamte Bildschirm ausgefüllt ist. Network Groups: Gruppiert die Controls nach Netzwerk und fasst sie auf einem Bildschirm zusammen. Die Gruppierung beginnt auf der definierten Ebene (0 = Alle, 1 = Erste Ebene...)
Dashboard Theme	Legt das Thema des HMI Dashboards fest, beeinflusst Steuerelemente und Hintergründe: Light: Helles Thema, insbesondere für den Tagmodus. Shiny: Ähnlich wie das helle Thema, Farbverlauf in den Controls. Dark: Dunkles Thema, insbesondere für den Nachtmodus.
Select Color	Legt die Farbe des Dashboard-Headers mit Farbverlauf fest, falls aktiviert.
Control Style	Legt den Stil der Controls fest: Round: Die Controls besitzen abgerundete Ecken. Flat: Die Controls besitzen kantige Ecken.
Change default background image	Falls aktiviert, kann ein kundenspezifisches Hintergrundbild für das HMI Dashboard festgelegt werden. Wenn nichts festgelegt wird, wird das Standardbild verwendet.
Use Logo	Falls aktiviert, wird dem Dashboard-Header ein Logo hinzugefügt. Es kann ein kundenspezifisches Bild für das Logo festgelegt werden.
Use custom Map Icon	Falls aktiviert, kann ein kundenspezifisches Karten-Icon für die Karte auf der Startseite festgelegt werden. Wenn nichts festgelegt wird, wird das Standard-Icon verwendet.

**Languages**

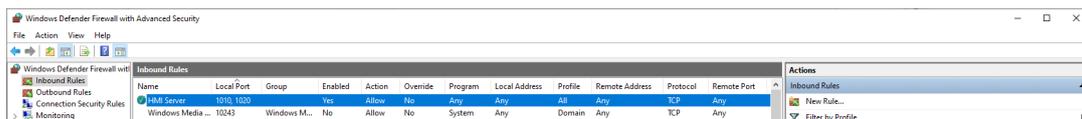
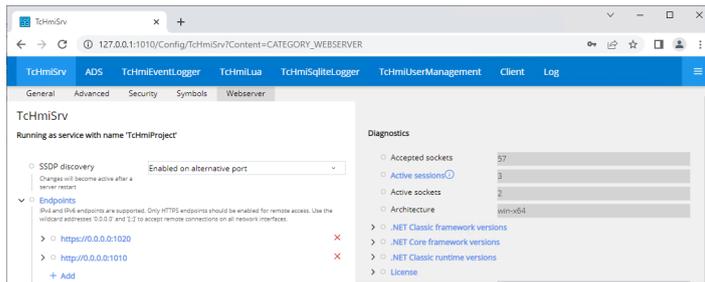
Languages	Aktiviert die aktivierte Sprache für den Sprachenwechsel im HMI Dashboard. (Nur für Standardtext)
-----------	---

**HMI Server**

<u>Publish to TwinCAT HMI Server</u>	Veröffentlicht das Dashboard an einen TwinCAT HMI Server, falls aktiviert.
Address	Geben Sie die IP-Adresse oder den Hostnamen des TwinCAT HMI Servers ein.
Port	Geben Sie den Port des TwinCAT HMI Servers ein (standardmäßig 1010 ohne Verschlüsselung und 1020 mit Verschlüsselung).
User	Geben Sie den Benutzernamen ein.
Password	Geben Sie das Passwort ein. (Das Administratorpasswort muss auf dem Server über die Konfigurationsseite festgelegt werden.)
Validate Connection	Drücken Sie die Schaltfläche, um Ihre Serververbindung zu validieren.

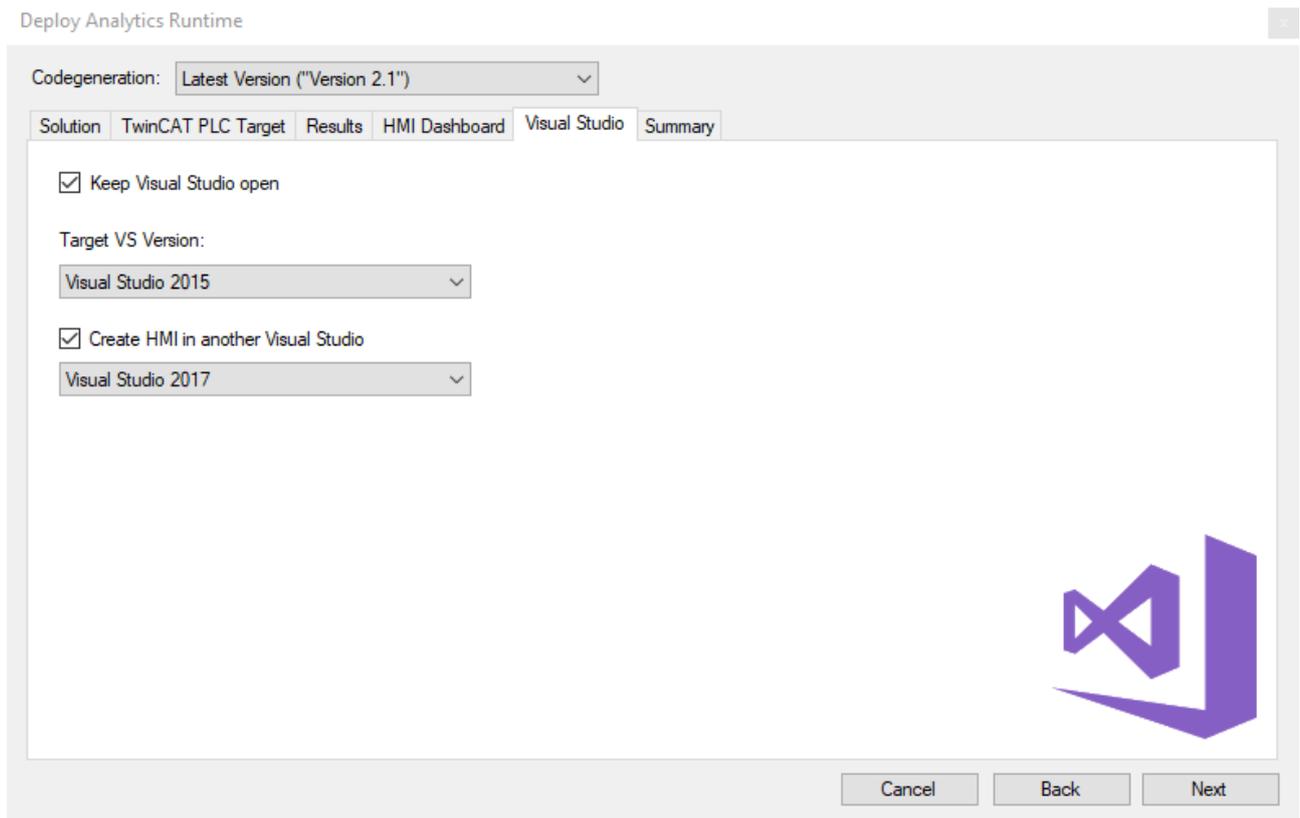


Damit zu einem Remote HMI Server veröffentlicht werden kann, muss der HMI Port und die IP des Entwicklungsrechners im HMI Server freigegeben werden. Ebenfalls muss eine Inbound Rule mit den HMI Ports in der Windows Firewall vom Remote-PC eingerichtet werden.



### Karteireiter Visual Studio

Auf dem Karteireiter **Visual Studio** kann ausgewählt werden, welche Visual Studio®-Version oder TwinCAT XAE Shell für die Generierung von SPS und HMI verwendet werden soll. Zudem besteht die Möglichkeit, HMI und SPS in zwei verschiedene Lösungen zu generieren.

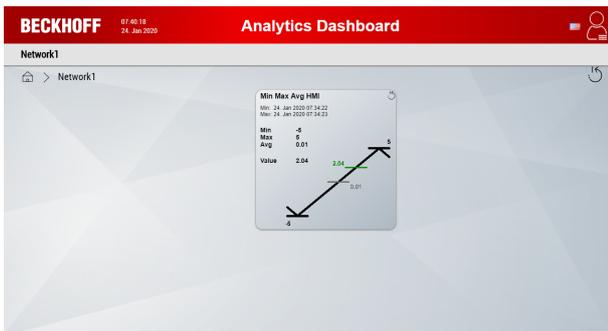
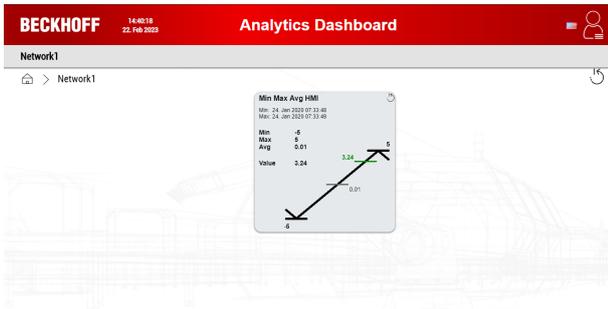
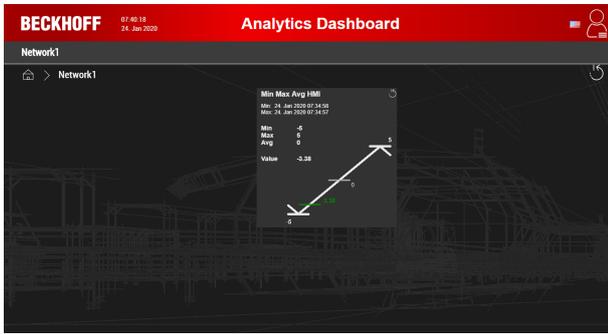


Keep Visual Studio open	Lässt Visual Studio® nach der Generierung geöffnet, falls aktiviert.
Target VS Version	Die Visual Studio®-Zielversion für SPS und HMI.
Create HMI in another Visual Studio	Falls aktiviert, wird die HMI in eine zweite Lösung generiert. Es kann eine andere Version von Visual Studio® ausgewählt werden.
Anforderung: TwinCAT bzw. TwinCAT HMI muss in der ausgewählten Visual Studio®-Version installiert sein.	

Klicken Sie auf **Next**, um den Karteireiter **Summary** anzuzeigen, und anschließend auf **Deploy**, um die Generierung des Dashboards zu starten.

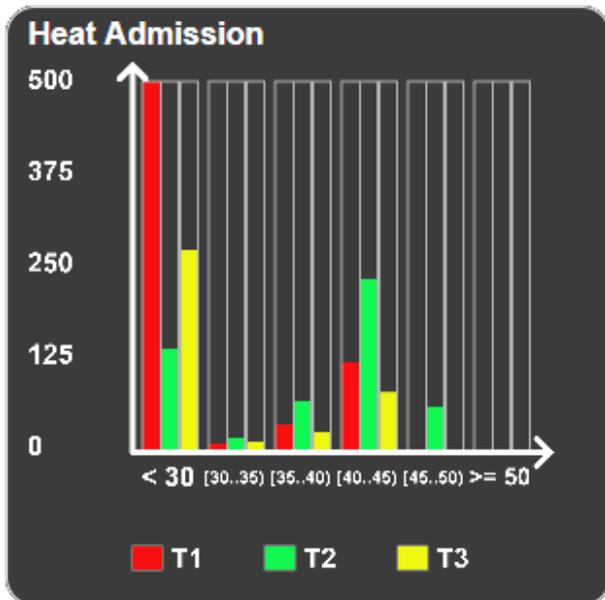
Impressionen

Themen

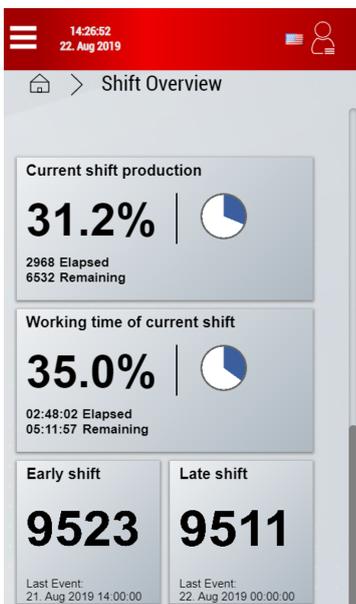


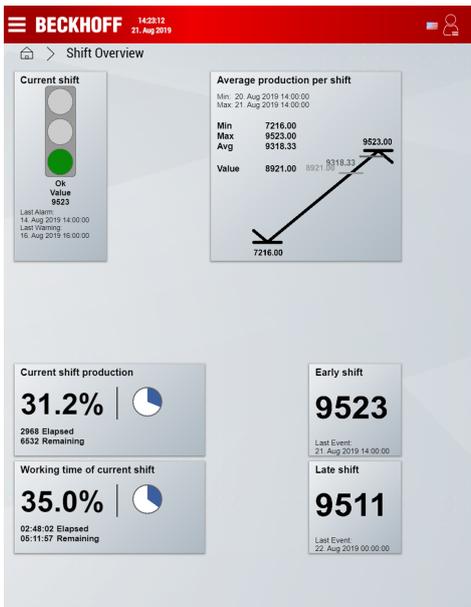
Stile



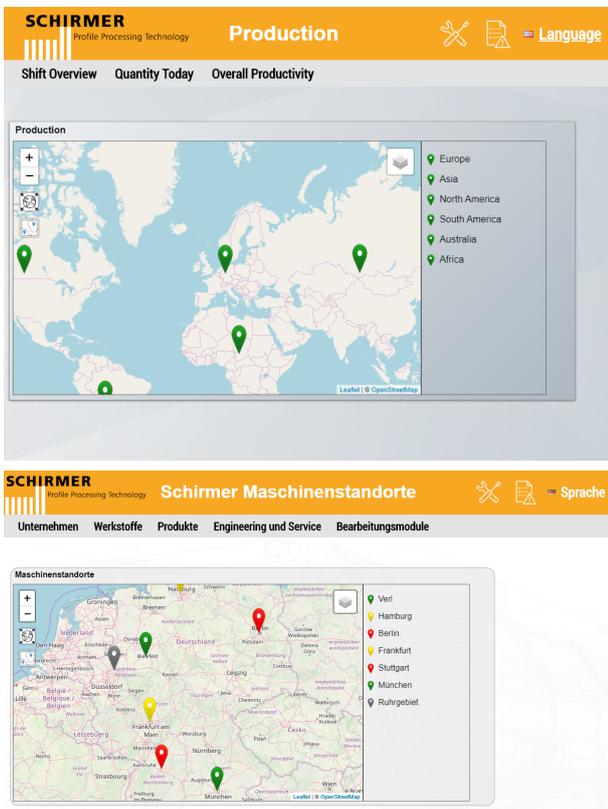


**Ansichten für Mobilgeräte**

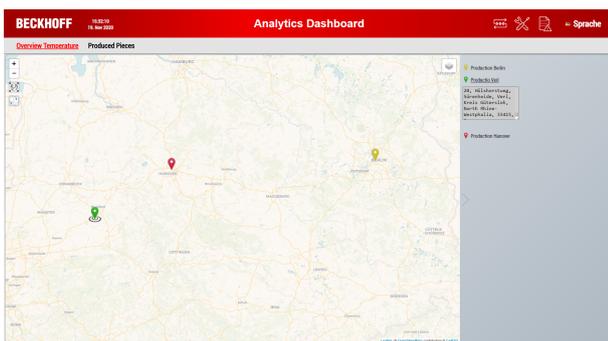


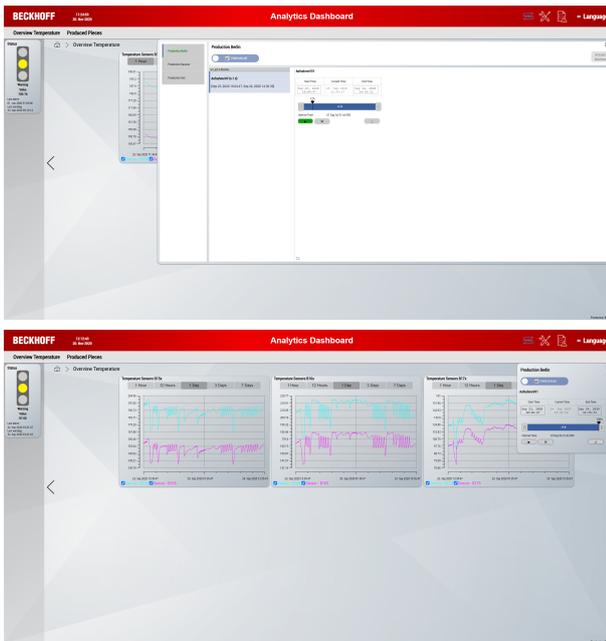


**Map auf kundenspezifischem Dashboard**



**Historische Daten und Maschinenumschaltung**





## 6.2.5 Ändern eines generierten Dashboards im HMI-Engineering

### TwinCAT 3 HMI-Projekt

Das Ergebnis der Dashboard-Generierung ist ein vollständiges TwinCAT 3 HMI-Projekt. Daher können alle Möglichkeiten, die das [TwinCAT 3 HMI Engineering](#) bietet, genutzt und einbezogen werden.

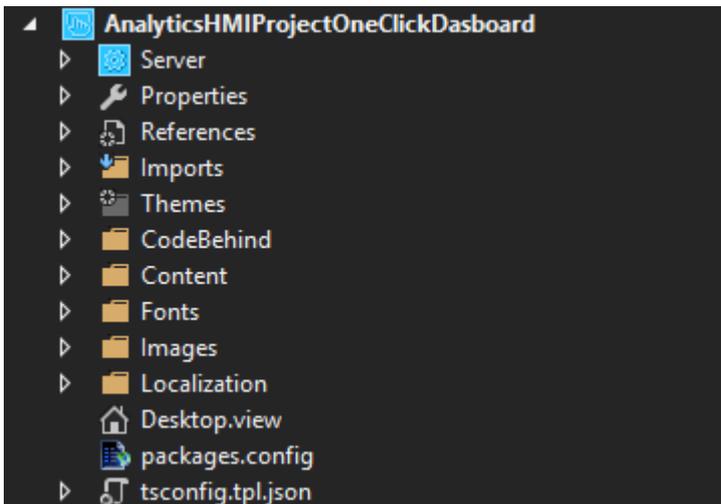


Abb. 1:



Beachten Sie, dass viele Dashboard-Anpassungen auch im [Webbrowser](#) zur Laufzeit vorgenommen werden können.

## 6.2.6 Konfiguration des Dashboards zur Laufzeit im Webbrowser

Zur Laufzeit können im Webbrowser unterschiedliche Konfigurationen vorgenommen werden:

- Erstellen individueller benutzerspezifischer Layouts
- Verwalten von Benutzern (siehe Benutzerverwaltung und Zugriffsrechte konfigurieren)
- Anpassen der Zugriffsrechte auf Contents (siehe Benutzerverwaltung und Zugriffsrechte konfigurieren)
- Ändern von Control-Eigenschaften

- Anpassen von Parametern
- Ändern von globalen Dashboard-Optionen



Beachten Sie, dass bei der Dashboard-Version 1.0 nicht alle Konfigurationsmöglichkeiten verfügbar sind.

### 6.2.6.1 Anpassen von Layouts mit der Interactive-Funktion

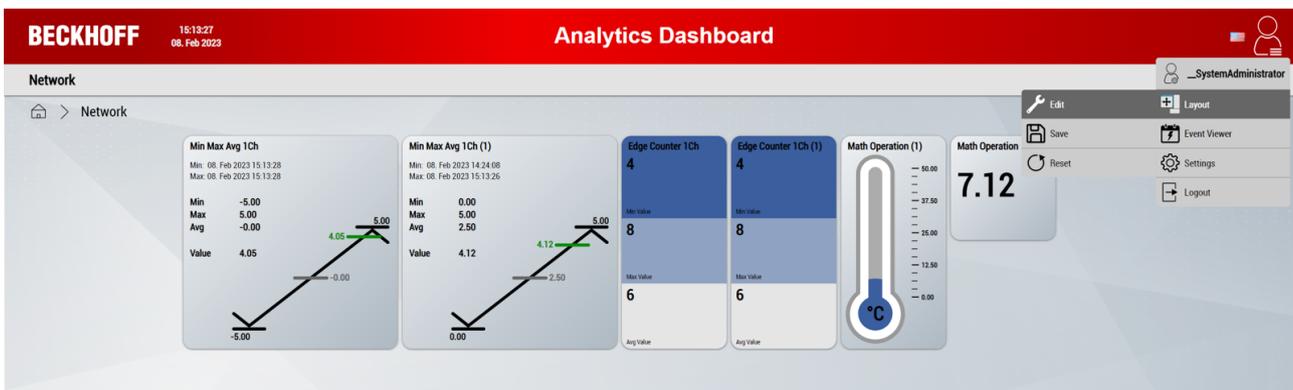
Mit der Interactive-Funktion lassen sich Controls innerhalb eines Contents individuell benutzerspezifisch anordnen. Das angepasste Layout wird zentral in der HMI-Server-Extension „TchmiAnalytics“ abgespeichert.



Beachten Sie, dass die Bearbeitung von Layouts erst ab der Dashboard-Version 2.0 verfügbar ist. Zudem kann das Layout nur dann bearbeitet werden, wenn der Benutzer über die notwendigen Zugriffsrechte verfügt.

#### Bearbeitungsmodus

Die Bearbeitung eines Layouts geschieht in einem speziellen Bearbeitungsmodus. Der Bearbeitungsmodus kann über das Menü-Control aufgerufen werden. Alternativ kann der Bearbeitungsmodus auch mit der Tastenkombination „STRG“ + „E“ gestartet werden.



Im Bearbeitungsmodus ist im Hintergrund ein Gitternetz sichtbar, auf welchem die Controls beliebig angeordnet werden können.



#### Layouts bearbeiten

Folgende Möglichkeiten stehen zur Verfügung:

- **Positionieren:** Für die Positionierung muss auf ein Control geklickt bzw. gedrückt werden. So wird es aus seiner Verankerung gelöst und ist verschiebbar. Der hellblaue Rahmen zeigt die neue Position innerhalb des Layouts an. Lässt man das Control los, verankert sich dieses an seiner neuen Position.

- **Skalieren:** Mithilfe des Pfeils in der unteren rechten Ecke eines Controls kann dessen Größe verändert werden.
- **Controls ausblenden:** Durch den Sichtbarkeits-Button (unten rechts innerhalb eines Controls) kann dieses ein- und ausgeblendet werden. Ausgeblendete Controls werden im Bearbeitungsmodus halbtransparent dargestellt.



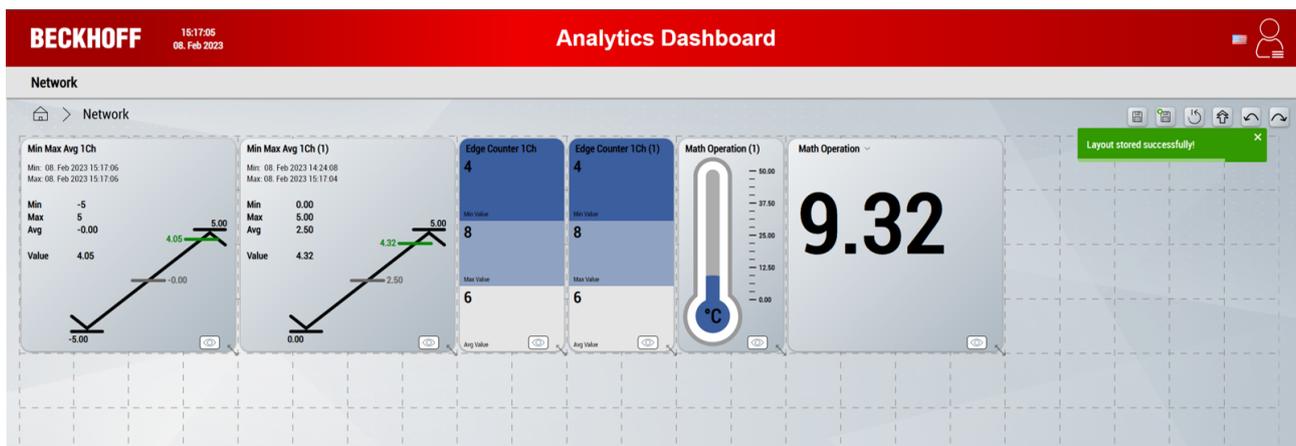
## Bearbeitungsmenü

Im Bearbeitungsmodus können über ein zusätzliches Menü (rot umrandet) unterschiedliche Aktionen vorgenommen werden:

- Speichern des Layouts (alternativ Tastenkombination „STRG“ + „S“)
- Speichern des Layouts und Verlassen des Bearbeitungsmodus (alternativ Tastenkombination „STRG“ + „Q“)
- Zurücksetzen der durchgeführten Änderungen (alternativ Tastenkombination „STRG“ + „R“)
- Controls nach oben verschieben, sodass freie Zeilen entfernt werden (alternativ Tastenkombination „STRG“ + „U“)
- Letzte Aktion rückgängig machen (alternativ Tastenkombination „STRG“ + „Z“)
- Letzte Aktion wiederholen (alternativ Tastenkombination „STRG“ + „Y“)

## Layouts speichern

Angepasste Layouts können über das Bearbeitungsmenü oder über das Menü-Control gespeichert werden. Ein Popup zeigt das Ergebnis des Speichervorgangs.



Wenn ein Fehler beim Speichern auftritt, prüfen Sie in der [Server-Konfiguration](#), ob die Extension „TchMiAnalytics“ aktiviert ist.

**Mobiles Layout**

Für mobile Geräte kann zusätzlich ein benutzerspezifisches mobiles Layout generiert werden. Dieses ist unabhängig von dem normalen Layout und besitzt lediglich 2 Spalten. Die Bearbeitung und das Speichern erfolgt auf dieselbe Art.



**6.2.6.2 Ändern von Control-Eigenschaften im Browser**

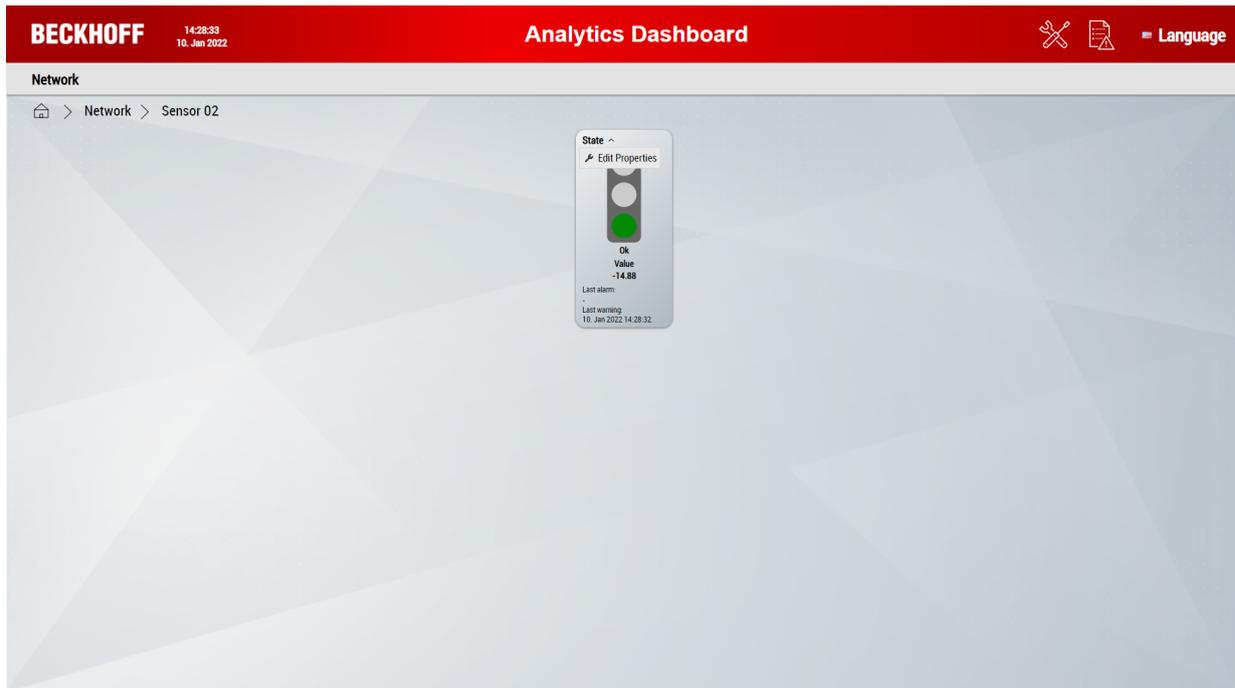
**Empfehlung: Änderungen im Analytics Workbench Projekt durchführen**

In der Regel können die hier vorgenommenen Änderungen bereits in der Analytics Workbench vorgenommen werden, bevor Sie das Dashboard generieren (siehe Mapping Wizard oder auch Dashboard-Knoten). Dies hat den Vorteil, dass die Änderungen bei einer erneuten Dashboard-Generierung gespeichert bleiben.

Jedes Analytics Control besitzt ausgewählte Eigenschaften, die dynamisch im Browser geändert werden können. Diese Möglichkeit ist ab der Analytics Version 3.4.3145.0 bzw. mit der Control Paket Version 1.1.31 verfügbar. Sie können damit bei älteren HMI Projekten einfach das NuGet Paket aktualisieren.

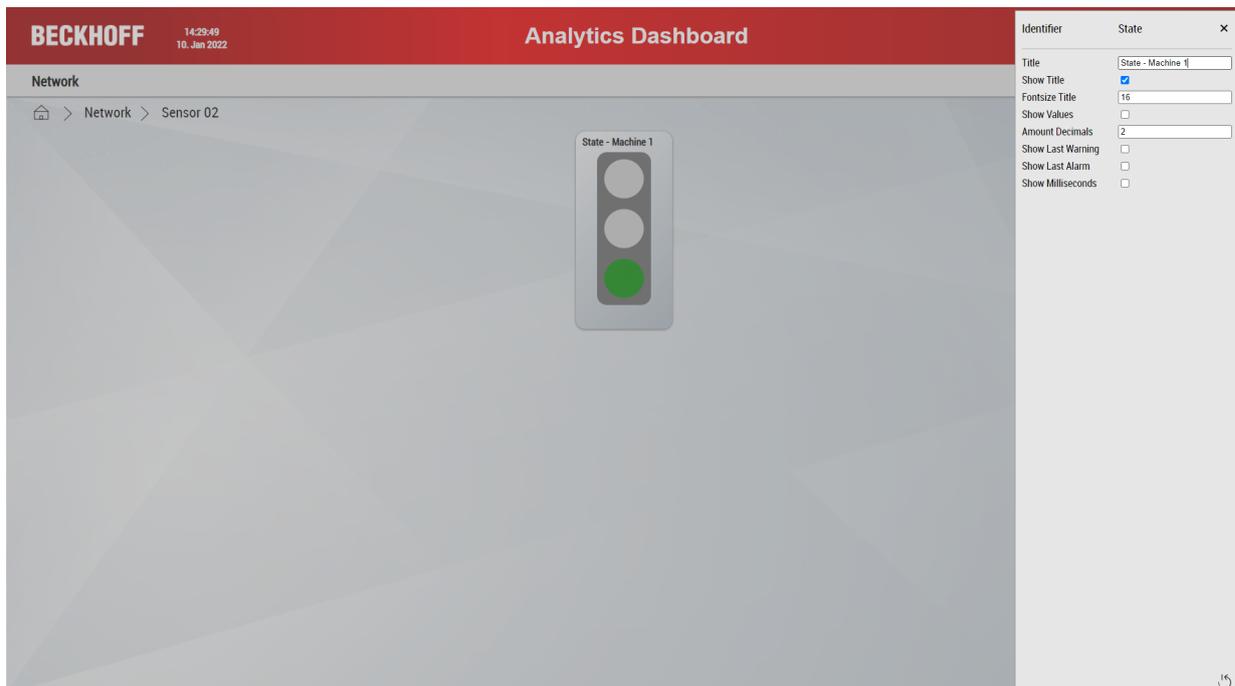
1. Öffnen Sie ein generiertes HMI Projekt und klicken Sie auf den Titel eines Controls, bei dem Sie eine Eigenschaft ändern wollen.

2. Dort klappt sich ein Menü aus, bei dem Sie auf **Edit Properties** klicken.



⇒ Jedes Control besitzt unterschiedliche Eigenschaften, die angepasst werden können. Hier werden zu Demonstrationszwecken die Eigenschaften des *Traffic Light Controls* geändert

3. Ändern Sie den Titel und haken die unteren drei Auswahlfelder ab, um lediglich die Ampel mit dem geänderten Titel anzuzeigen.

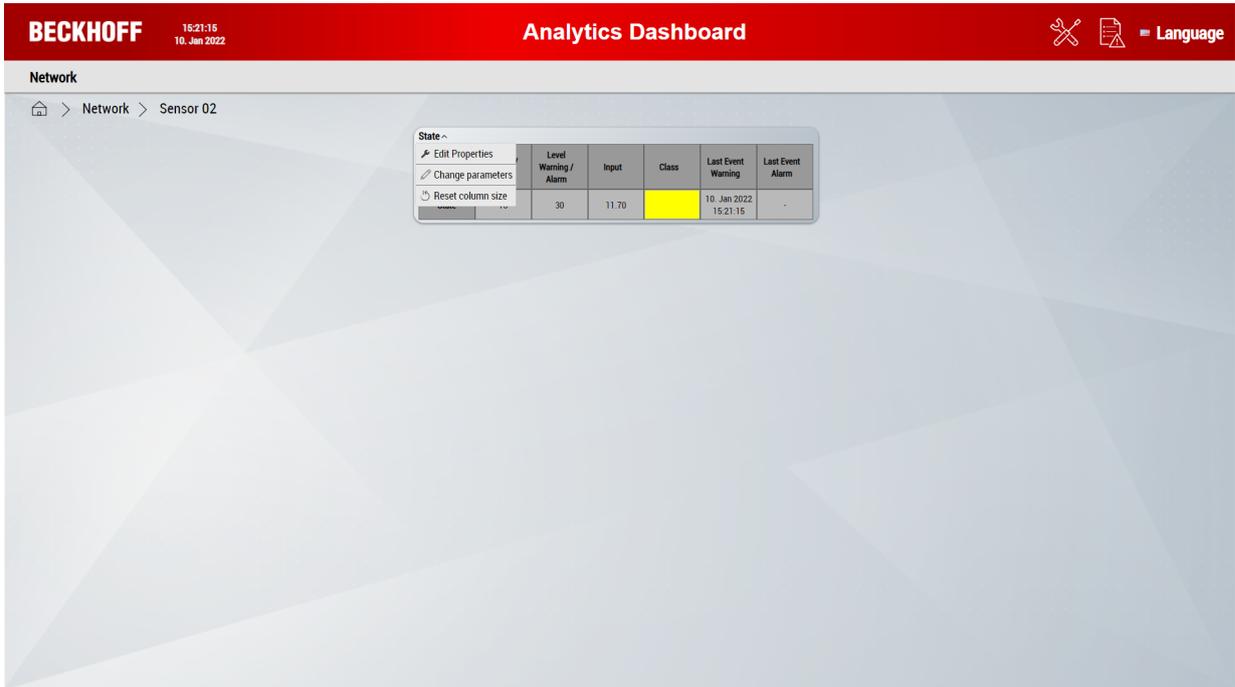


⇒ Die vorgenommenen Änderungen werden Client-Seitig (auf dem Gerät, auf dem Sie das Dashboard öffnen) gespeichert und lediglich dort angezeigt. Zukünftig wird die Speicherung über den HMI Server durchgeführt, wodurch Änderungen global auf jedem Gerät durchgeführt werden.

### 6.2.6.3 Ändern von Parametern im Browser

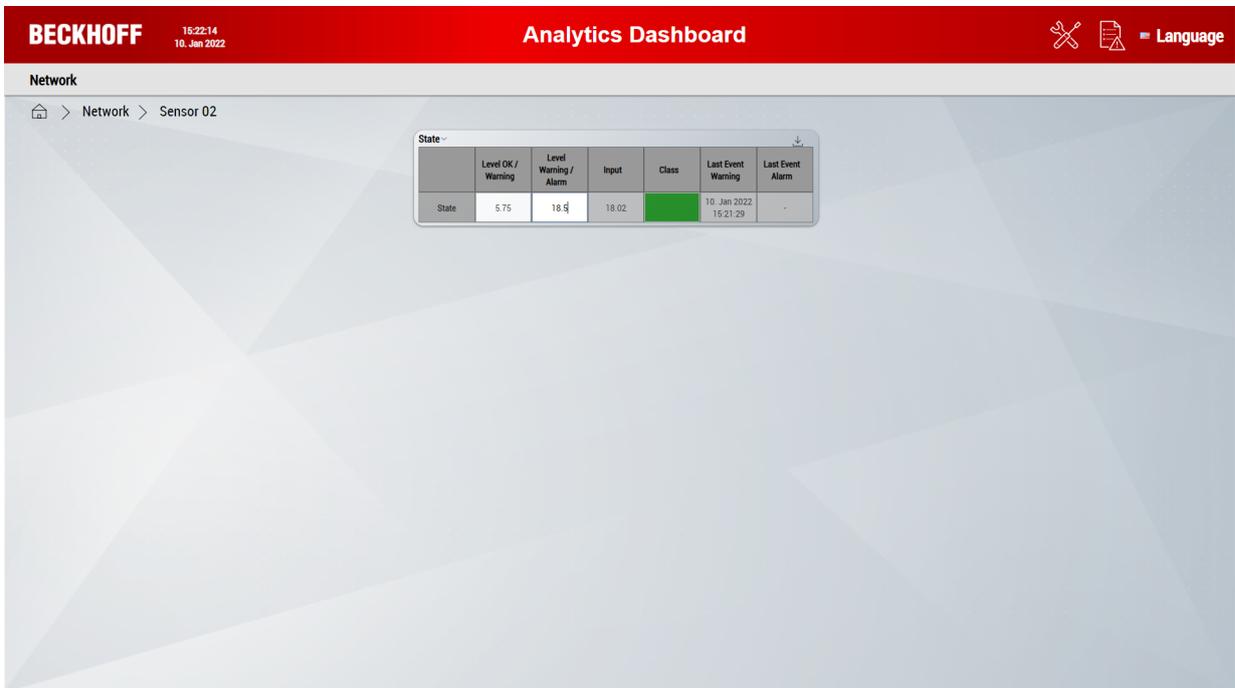
Numerische Parameter wie Grenzwerte können dynamisch im Dashboard geändert werden. Dies ist mit dem Data Table Control möglich. Die Werte werden dort persistent zurück in die SPS geschrieben und sind damit auch nach einem Neustart verfügbar.

1. Öffnen Sie ein generiertes HMI Projekt und klicken Sie auf den Titel des Tabellen Controls, bei dem Sie ein Parameter ändern wollen.
2. Dort klappt sich ein Menü aus, bei dem Sie auf **Change parameters** klicken. Ein Modul ohne Parameter hat diesen Menüeintrag nicht.

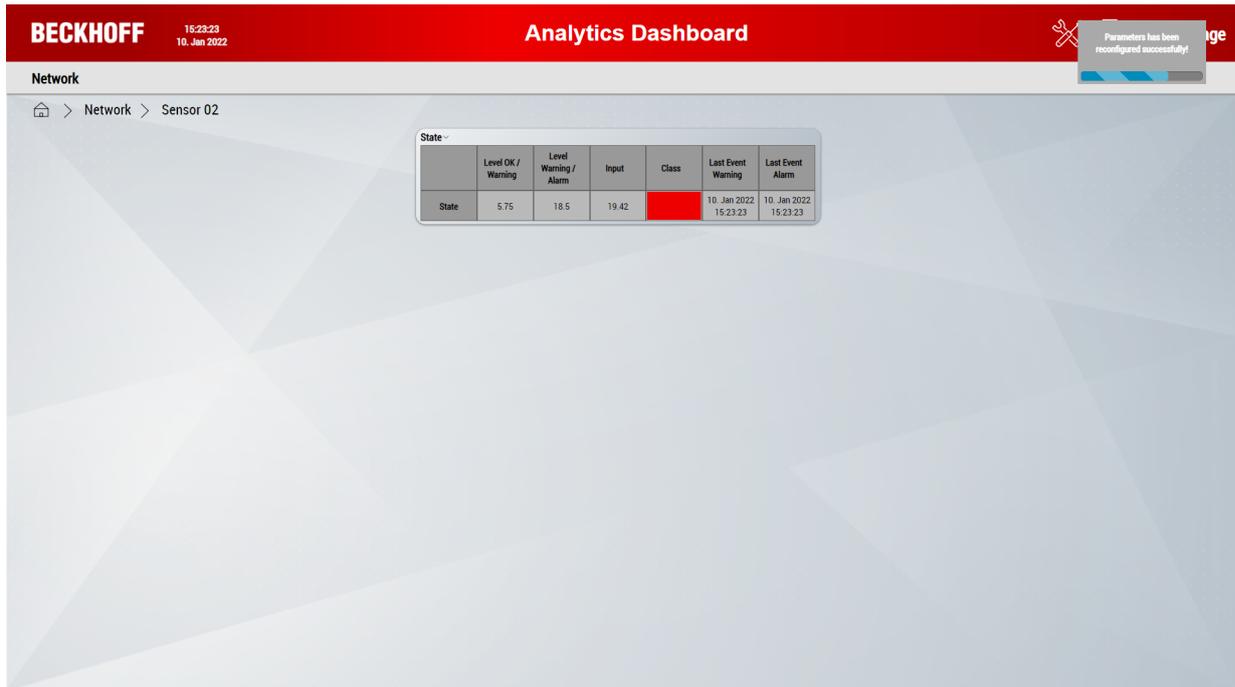


⇒ Die änderbaren Felder wechseln zu einem Eingabefeld mit weißem Hintergrund.

3. Ändern Sie dort die Werte und klicken Sie anschließend oben rechts in der Ecke auf das Speichern-Symbol.



- ⇒ Nach dem Speichern der Parameter wird Ihnen eine Meldung im oberen rechten Bereich des Dashboards angezeigt. Dort wird Ihnen mitgeteilt, ob die Neukonfiguration funktioniert hat.

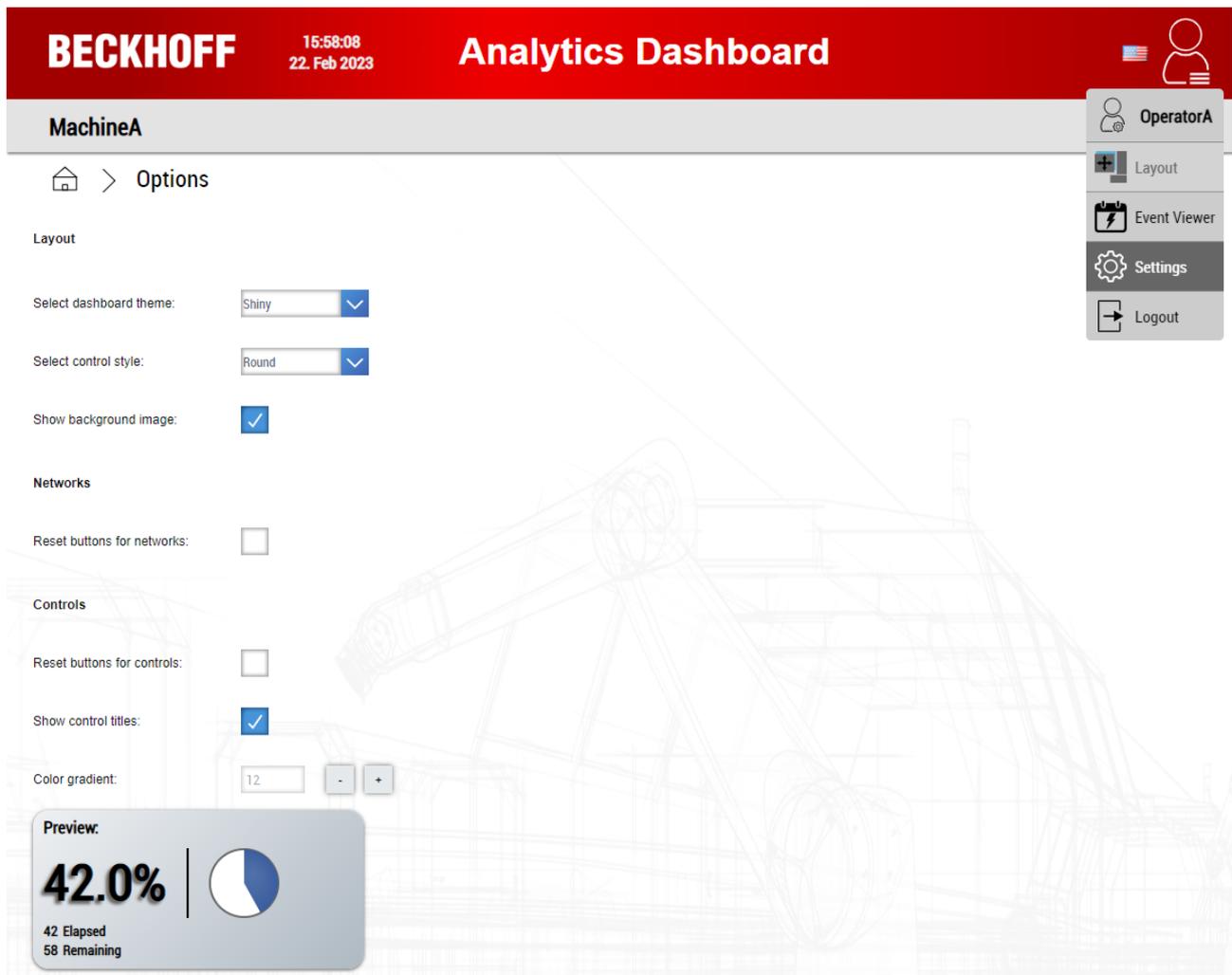


The screenshot shows the Beckhoff Analytics Dashboard interface. At the top, there is a red header with the Beckhoff logo, the time '16:23:23', the date '10. Jan 2022', and the title 'Analytics Dashboard'. A notification box in the top right corner states 'Parameters has been reconfigured successfully!'. Below the header, the breadcrumb navigation shows 'Network > Sensor 02'. A table titled 'State' displays the following data:

	Level OK / Warning	Level Warning / Alarm	Input	Class	Last Event Warning	Last Event Alarm
State	5.75	18.5	19.42		10. Jan 2022 15:23:23	10. Jan 2022 15:23:23

#### 6.2.6.4 Anpassen von globalen Dashboard-Optionen

Auf der Options-Seite können generelle Einstellungen für das Dashboard bearbeitet werden.



**Layout**

- Select dashboard theme
- Select control style
- Show background image

Ändern Sie das Dashboard-Thema zwischen Shiny, Dark und Light.  
 Ändern Sie den Steuerelementstil zwischen Flat und Round.  
 Zeigt das Standard- oder ein kundenspezifisches Hintergrundbild an.

**Networks**

- Reset buttons for networks

Aktiviert Reset-Schaltflächen für ganze Netzwerke. Zurücksetzen aller Controls innerhalb des ausgewählten Netzwerks.

**Controls**

- Reset buttons for controls
- Show control titles

Aktiviert Reset-Schaltflächen für Controls. Zurücksetzen eines einzelnen Controls.  
 Aktiviert Control-Titel für alle Controls.

## 6.2.7 Umschaltung mehrerer Maschinen im HMI Dashboard

In TwinCAT Analytics können Sie verschiedene Datenströme von mehreren Maschinen verwenden, die Sie in einer Analyse umschalten können. Dies geht sowohl in der Analytics Workbench über die Virtual Input Source als auch in der fertig generierten SPS und HMI. Dabei können sowohl live als auch historische Daten verwendet werden. Für jeden Datenstrom können Sie entweder im [Analytics Logger](#) oder in der [Machine Administration](#) [▶ 27] eine kurze Beschreibung und den Standort hinzufügen.

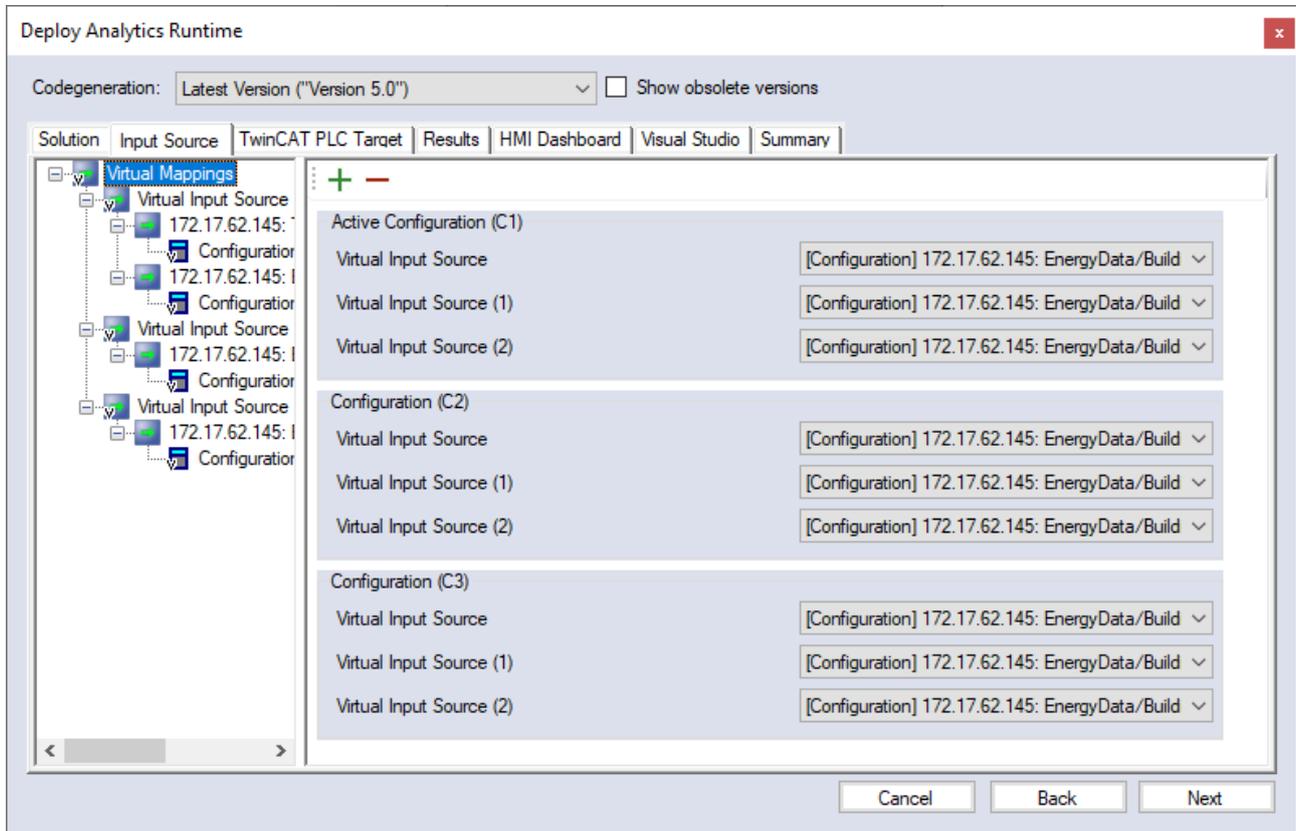
**Seite Machine Administration**

Falls nicht bereits in der Übersicht geschehen, legen Sie die Metadaten Ihrer Maschine auf der Seite [Analyse der Daten](#) [▶ 27] fest. Öffnen Sie den TwinCAT Target Browser (**TwinCAT > Target Browser > Target Browser**) und klicken Sie auf das Zahnrad-Symbol. Nun können Sie den Standort Ihrer Maschine, eine

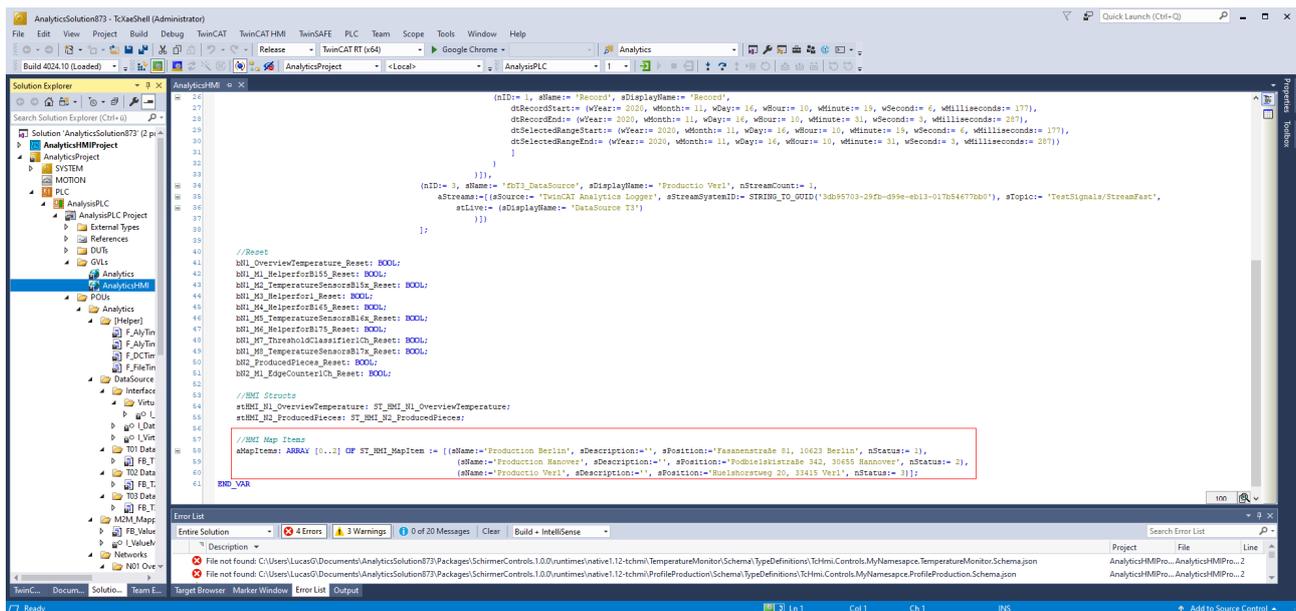


### Deploy Wizard

Im Runtime-Deployment [▶ 44] können unter **Input Source** beliebig viele Konfigurationen erstellt werden, die in dem HMI umschaltbar sind. Damit ist es in der Analyse HMI möglich, live sowie historische Daten von ggf. unterschiedlichen Maschinen zu analysieren und umzuschalten. Jede der aufgelisteten Konfigurationen können parallel analysiert werden. Eine Konfiguration besitzt immer so viele Virtual Input Sources, wie in der Analytics Workbench konfiguriert sind.

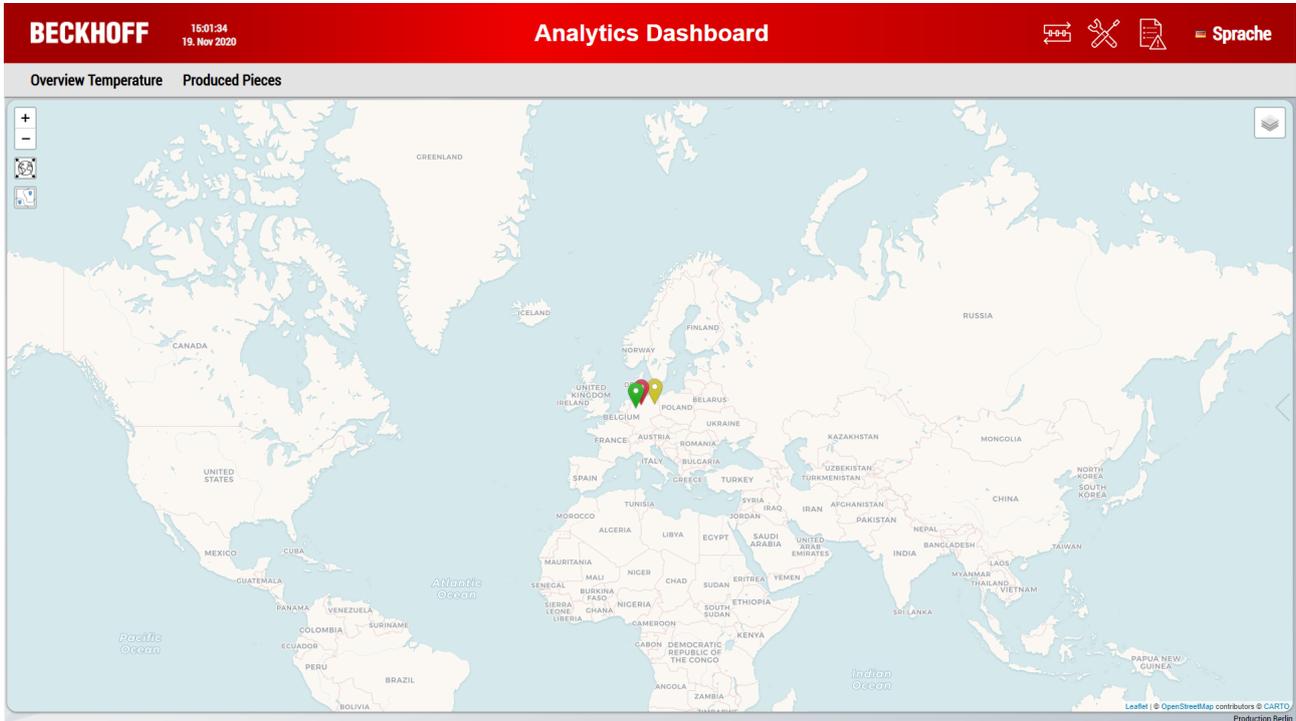


Nach einer erfolgreichen HMI und SPS Generierung sind die Namen, Positionen, und Beschreibungen der Input Sourcen in die globale Variablenliste der SPS eingetragen. Falls etwas nicht stimmig ist oder Sie einen Wert nachträglich ändern möchten, können Sie dies direkt in der SPS tun.

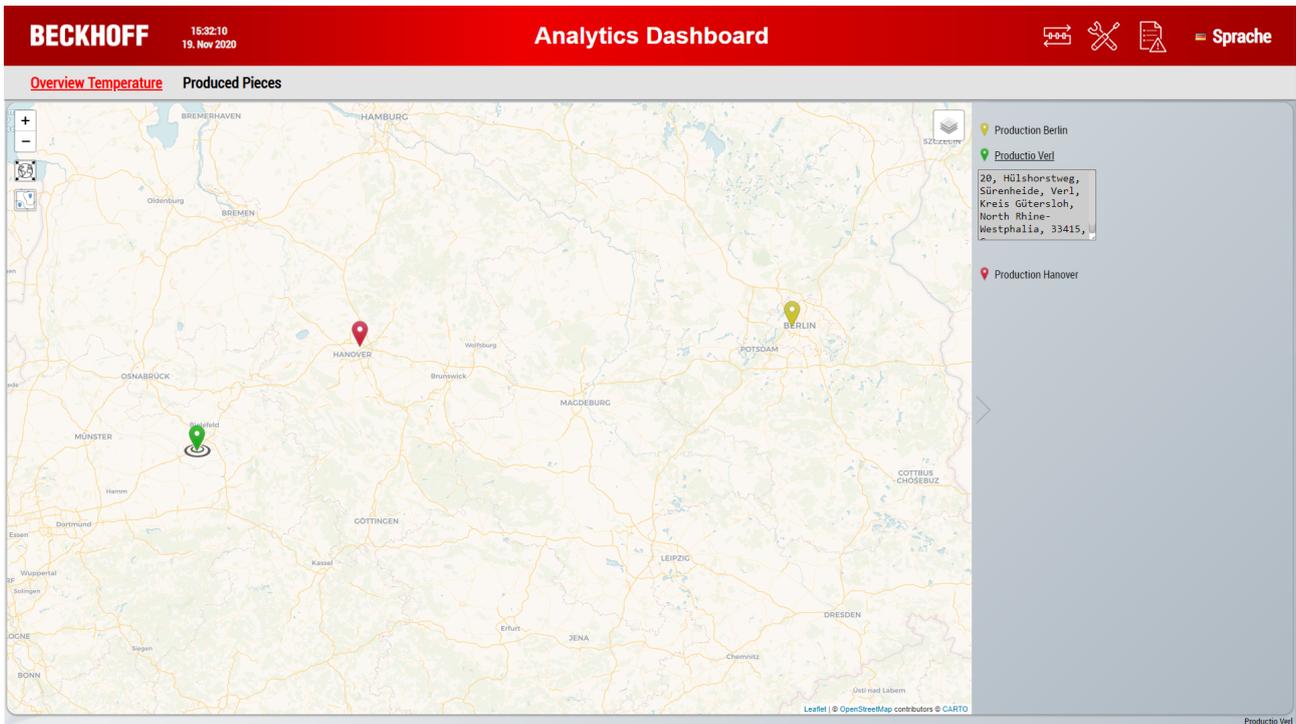


**Karte mit Maschinen**

Wenn Sie die Erstellung einer Startseite ausgewählt haben, wird das *Map Control* auf der Homepage Ihres HMI Dashboards erzeugt. Die Karte zeigt alle Maschinenstandorte an und listet die Namen der Maschinen (System Alias) in der Legende auf. Die Legende kann über den Pfeil rechts geöffnet und geschlossen werden. Die Farbe des Icons steht für den aktuellen Maschinenstatus: Grün = OK, Gelb = Warnung, Rot = Alarm.

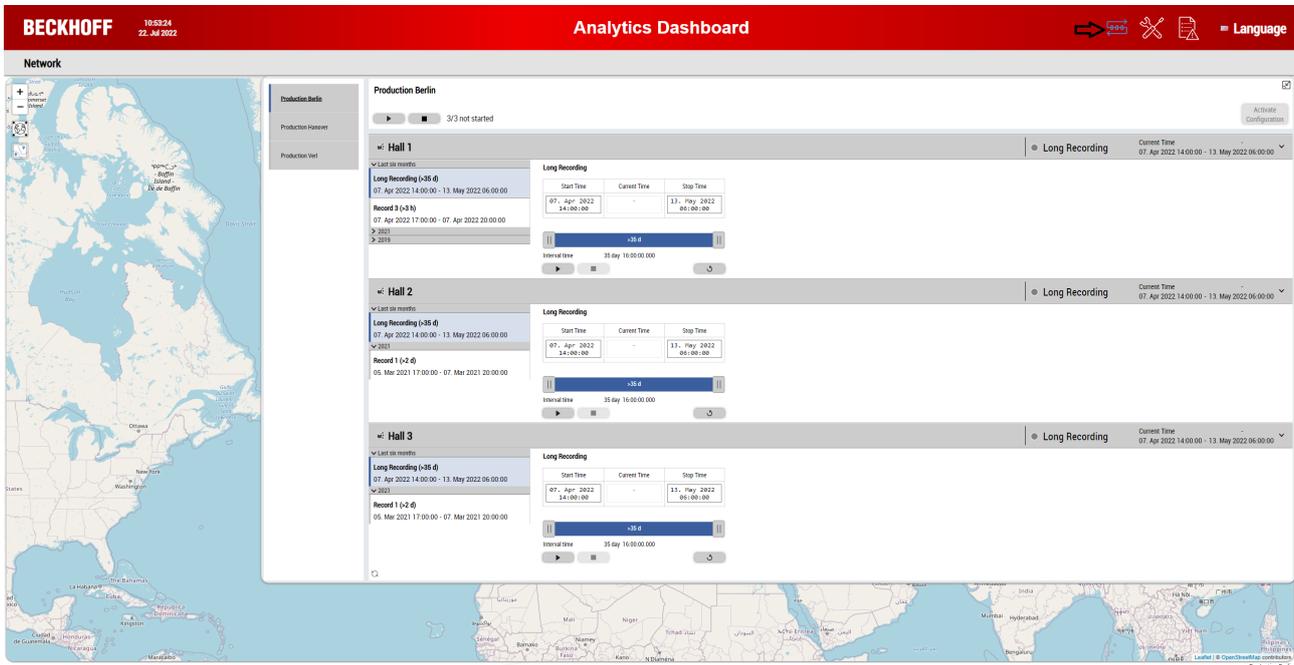


Sie können ein einzelnes Icon durch Doppelklick heranzoomen und mit den Icons auf der linken Seite auf eine Übersicht aller Maschinen zoomen. Zudem besteht die Möglichkeit, auf ein Icon in der Karte zu klicken. Der entsprechende Legendeneintrag wird automatisch hervorgehoben. Dies funktioniert auch andersherum. Wenn Sie auf einen Legendeneintrag klicken, wird das entsprechende Icon automatisch hervorgehoben.



## Umschaltung von Maschinendaten

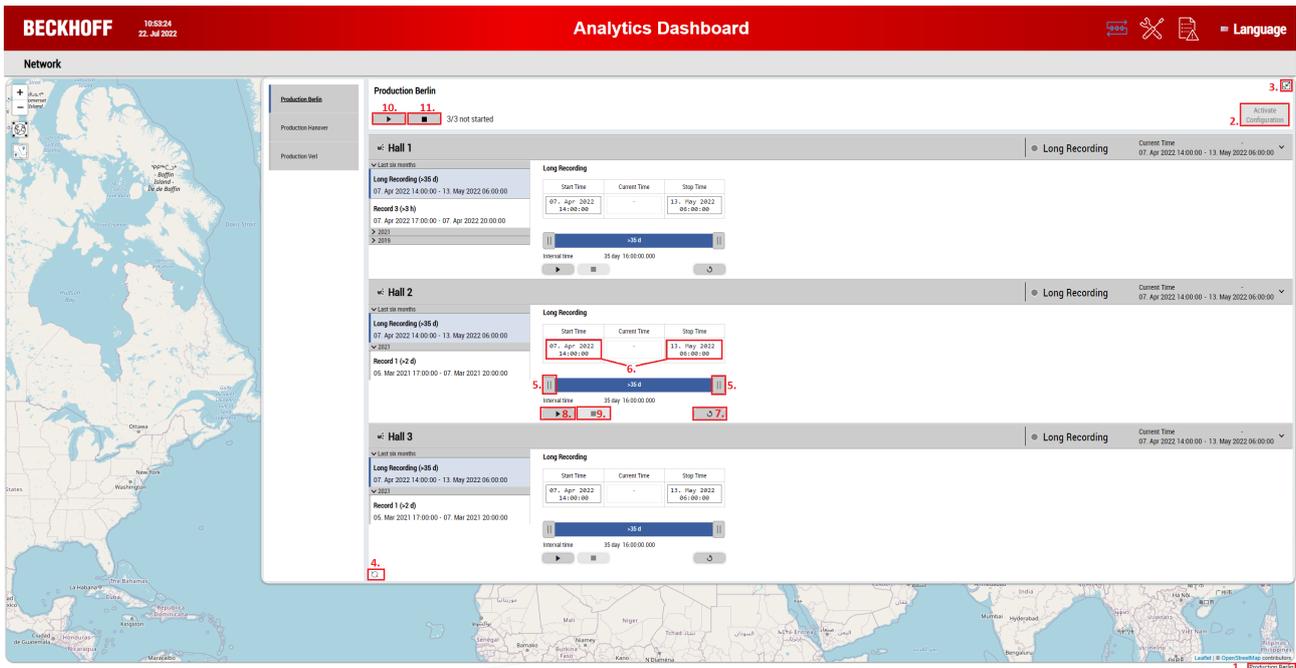
Neben der Karte mit den einzelnen Maschinenkonfigurationen lassen sich die Analysen zu den einzelnen Konfigurationen umschalten. In der SPS laufen alle Analysen in separaten Tasks, wodurch alle Analysedaten gleichzeitig zur Verfügung stehen. In dem generierten HMI können damit die verschiedenen Maschinen, in einem dafür speziell entwickelten Control, umgeschaltet werden. Das Control lässt sich über das im nachfolgenden Bild markierte Icon (schwarzer Pfeil) öffnen und schließen.



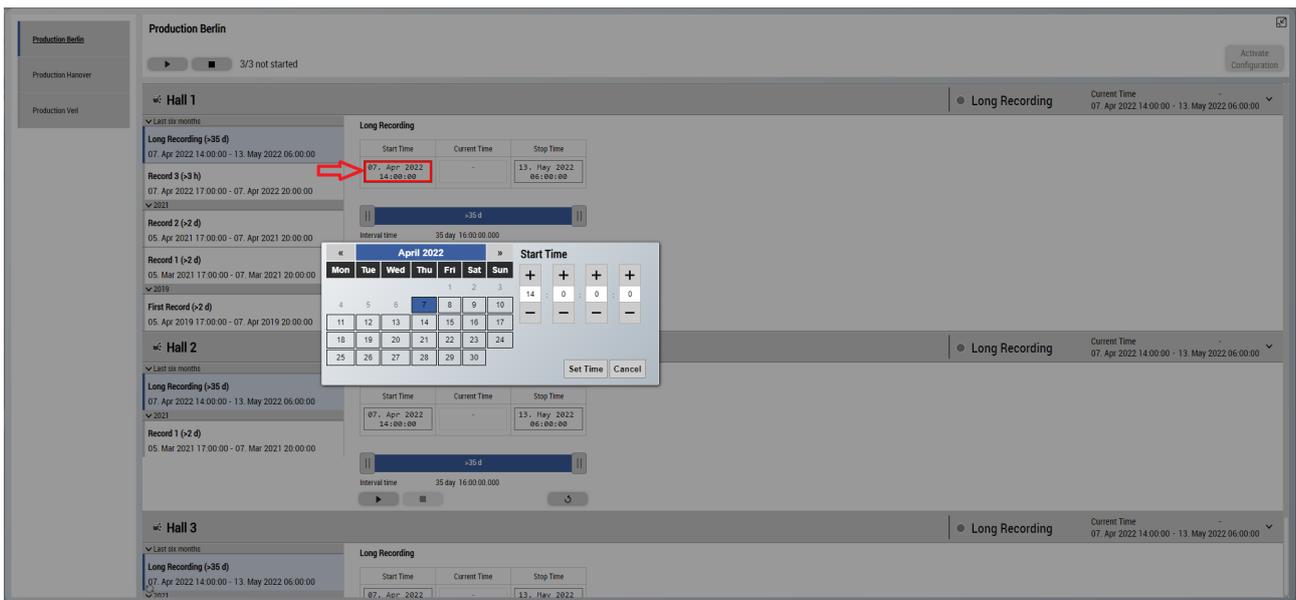
Die Konfigurationen, die zuvor auf der Karte mit den Standorten abgebildet waren, sind in dem Control auswählbar. Eine Konfiguration kann aus live und historischen Daten bestehen, wobei für die **Production Berlin** nur historische Daten konfiguriert sind. Historische Daten können äquivalent wie im Arbeit mit historischen Daten in der HMI analysiert werden.

Beim Klicken auf eine andere Maschine wird diese Konfiguration nur im Control angezeigt. Klicken Sie zum Aktivieren auf **Activate Configuration**. Im Folgenden wird das Control erklärt.

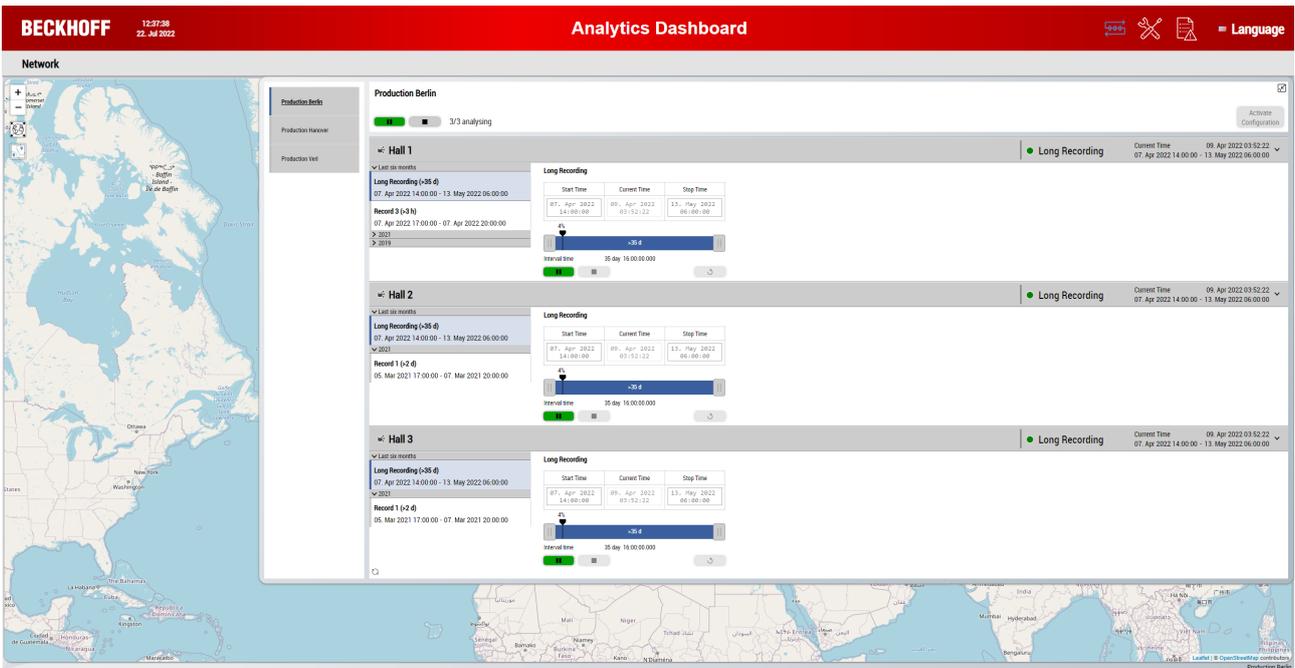
1. Die derzeit aktive Konfiguration.
2. Button, um die selektierte Konfiguration zu aktivieren.
3. Auswahlfenster zum Minimieren (und automatischer Aktivierung der selektierten Konfiguration).
4. Neuladen der Record-Liste. Damit können auch im Nachhinein aufgenommene Daten im Dashboard dynamisch geladen und analysiert werden.
5. Auswahl des Start- und Endzeitpunkts über einen Slider.
6. Auswahl des Start- und Enddatums über ein Auswahlfenster mit Kalender.
7. Zurücksetzen des Start- und Endzeitpunkts.
8. Starten der Analyse der selektierten **Input Source**.
9. Abbrechen der Analyse der selektierten **Input Source**.
10. Starten der gesamten Analyse
11. Abbrechen der gesamten Analyse



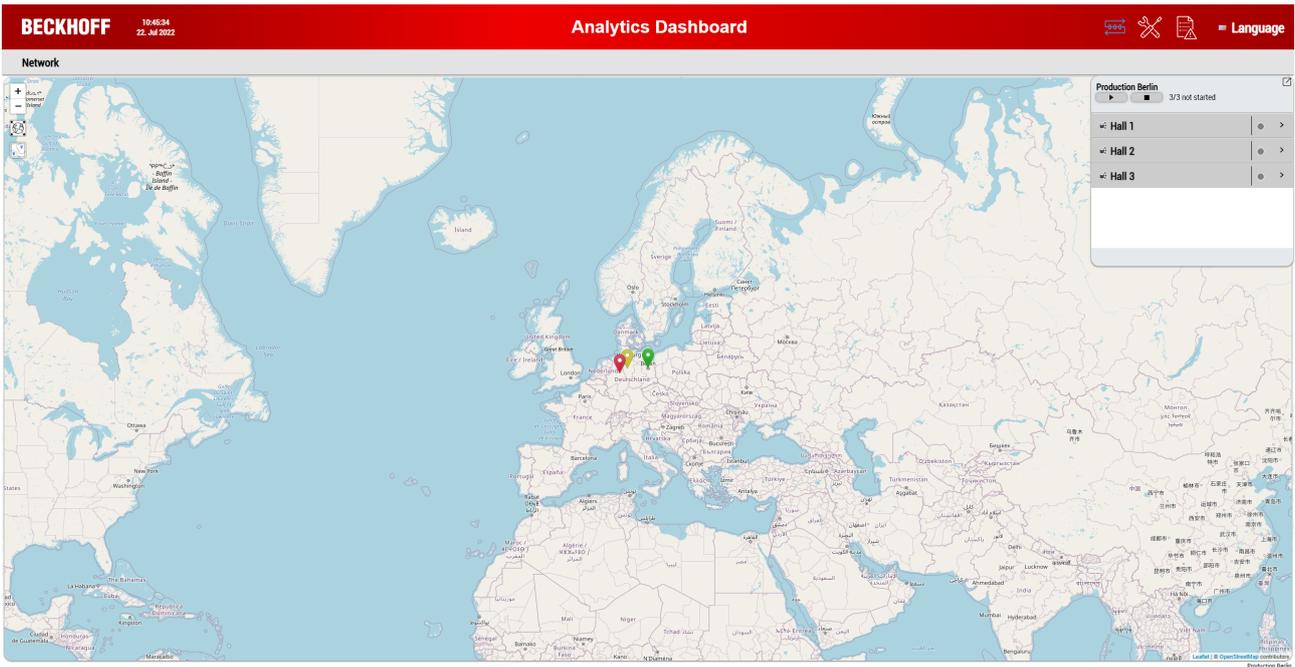
Für schnelle Tests und grobe Einstellung der Start- und Endzeit ist der Slider optimal. Klicken Sie für eine genaue Einstellung bis auf die Millisekunde auf das **Textfeld**. Damit öffnet sich ein Auswahlfenster mit Kalender zur Selektierung des Datums und zur Einstellung der genauen Uhrzeit.

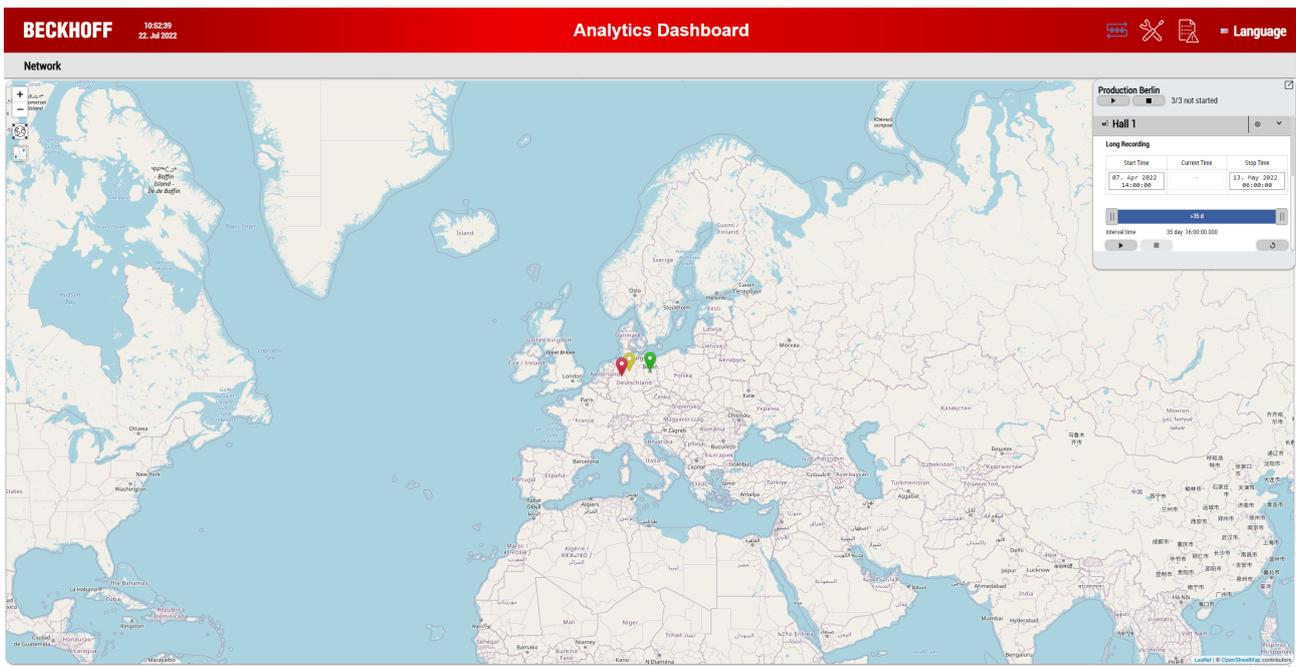


Sobald Sie den Button **Start** klicken, wird die Analyse mit allen Input Sources in der SPS angetriggert. Erstmals erscheint ein Ladesymbol, bis die Analyse anläuft. Dieser wird grün, sobald die ersten Daten erscheinen. Ebenfalls zeigt ein Marker den aktuellen Zeitpunkt der historischen Analyse im Slider an. Je nach Bandbreite, Anzahl der Daten und aufgenommene Zykluszeit kann der Prozess schneller oder langsamer abgearbeitet werden. Es ist möglich mehrere historische Aufnahmen gleichzeitig zu analysieren, indem einfach ein Maschinenwechsel vollzogen wird. Intern laufen alle Analysen parallel weiter. Im Dashboard kann damit einfach zwischen den Analysen umgeschaltet werden.



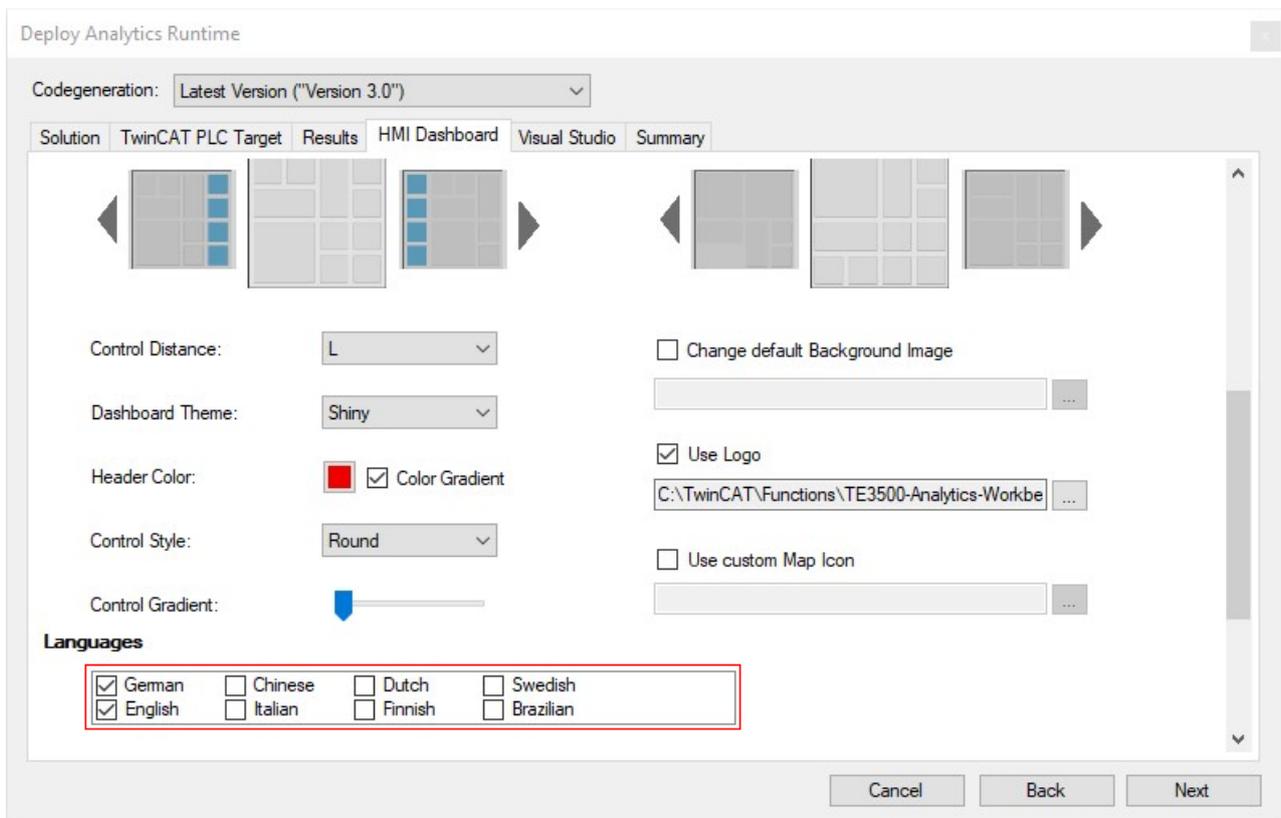
Über den Button **Minimieren** oben rechts in der Ecke wird die Auswahl der Konfigurationen und der Records ausgeblendet. Damit können Sie mit den Records von den Input Sourcen arbeiten und gleichzeitig die Daten betrachten. Sie können das Fenster ganz verstecken, indem Sie oben auf das blaue Icon klicken. Lediglich der Konfigurationsname steht immer rechts unten in der Ecke des Dashboards.





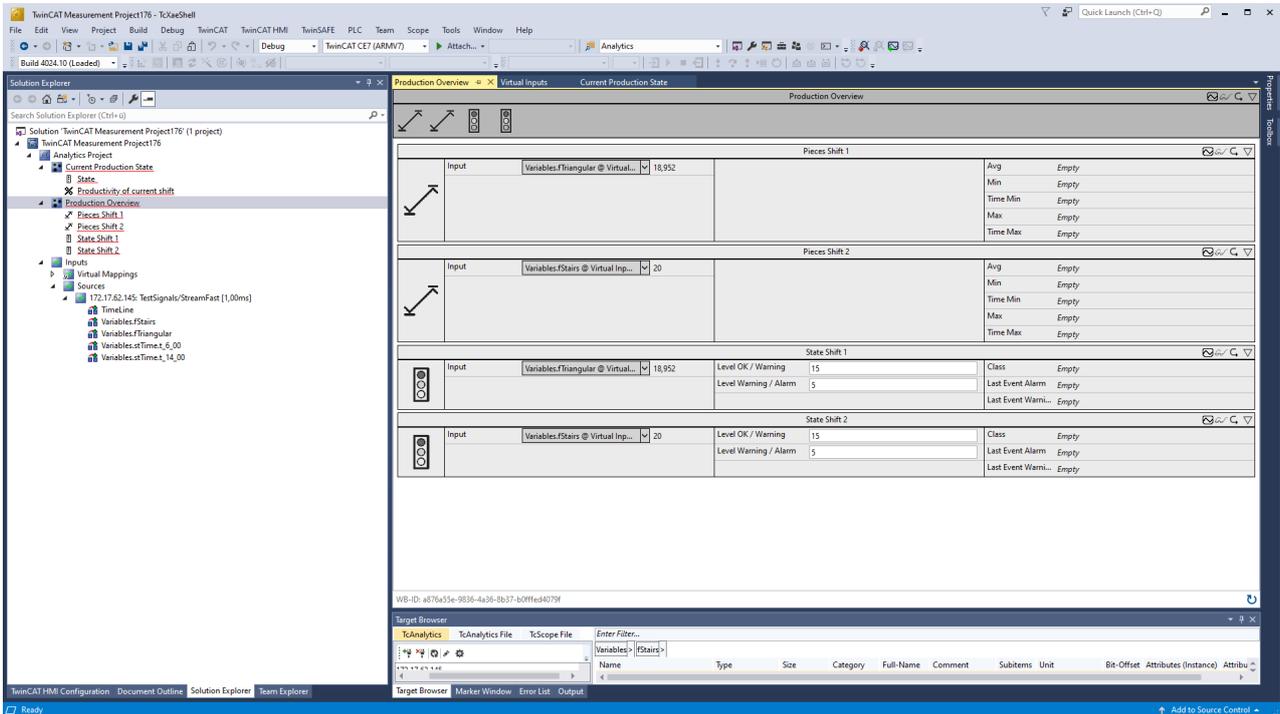
## 6.2.8 Integration einer Sprachumschaltung

In dem Konfigurationsfenster des Deploy Wizards können im Bereich der HMI bis zu 8 Sprachen für eine Sprachumschaltung ausgewählt werden.

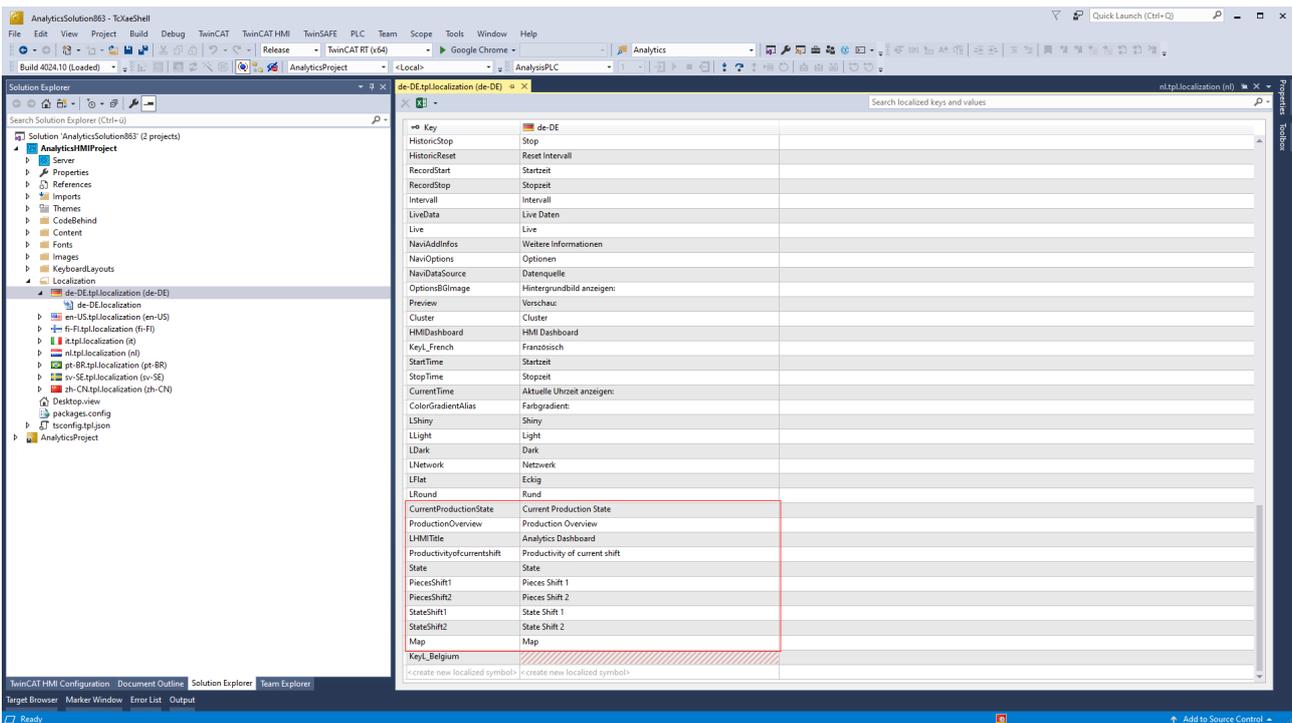


Wenn Sie die Benutzerverwaltung verwenden sind die Sprachen der angelegten Benutzer bereits ausgewählt. Diese können nicht abgewählt werden.

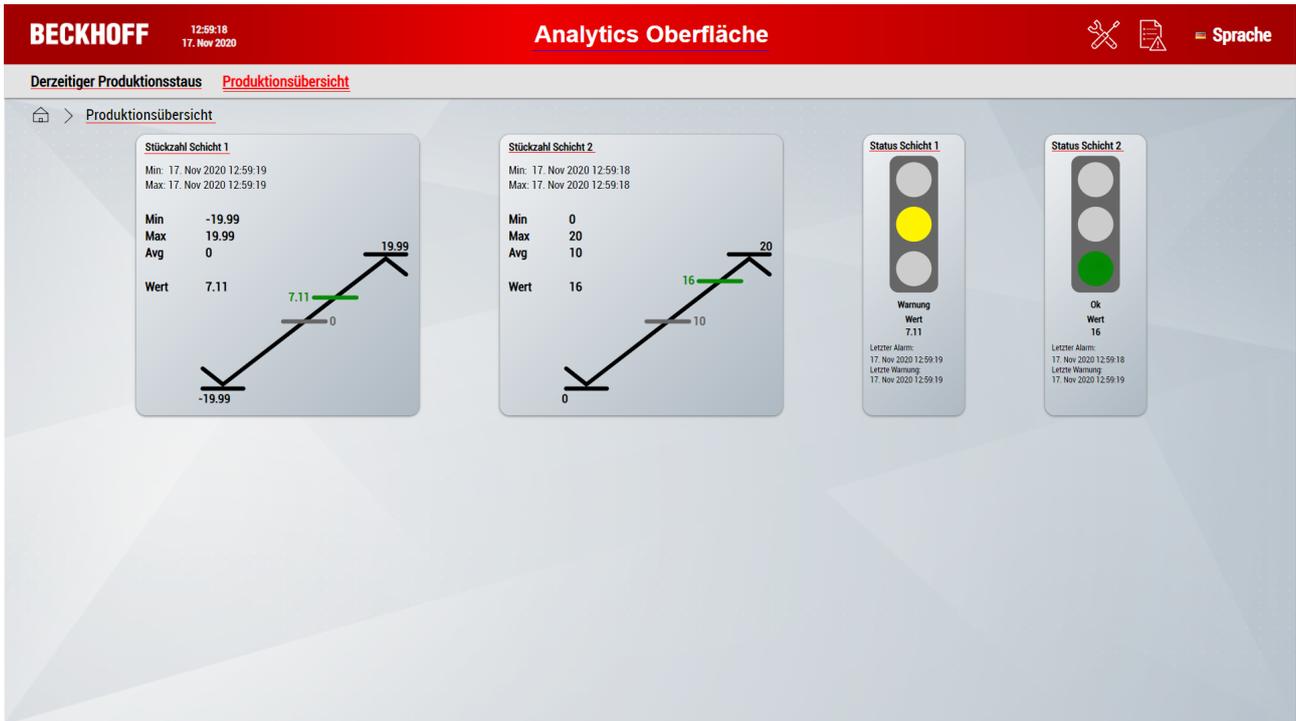
Die Texte in den mitgelieferten Analytics Controls sowie alle anderen Texte können automatisch umgeschaltet werden. Lediglich für die Benennung Ihrer Netzwerke und Module müssen Sie die Texte übersetzen, soweit Sie sich auch dafür eine Sprachumschaltung wünschen. In dem folgenden Screenshot sind die wesentlich betroffenen Namen im **Solution Explorer** gekennzeichnet.



In dem generierten TwinCAT HMI Engineering Projekt ist dies auf einem einfachen Weg möglich, da die Übersetzungseinträge schon vorbereitet sind. In den **Localization** Dateien werden die Texte für die jeweiligen Sprachen gespeichert. Die Namen Ihrer Netzwerke und Module werden automatisch dort eingetragen, es muss lediglich die Übersetzung vorgenommen werden. Dafür gehen Sie in alle Dateien der Sprachen, für die Sie eine Übersetzung benötigen. In dem folgenden Screenshot sind die Einträge aufgelistet, die für die deutsche Sprache übersetzt werden müssen. Die Anzahl der zu bearbeitenden Einträge variiert je nach Komplexität des Analytics Projekts.



Danach können Sie das Dashboard über den Button **Google Chrome** (bzw. der Name ihres Standardbrowsers) öffnen. Ein reines Neuladen im Browser hilft nicht, da intern das Projekt neu gebaut werden muss. Im folgenden Bild sind die Texte markiert, die nun zusätzlich mit der Sprachumschaltung umgeschaltet werden.



Über die Flagge in der oberen rechten Ecke kann die Sprache gewechselt werden.



Die Sprache kann ab der Dashboard-Version 2.0 benutzerspezifisch festgelegt und gespeichert werden.

## 6.3 Analytics Reporting

Auch in der Analytics Runtime ist ein 24/7-Reporting möglich. Über den TwinCAT Analytics Workbench Configurator lassen sich die dafür vorgesehenen Reporting Module konfigurieren (siehe 24/7-Reporting). Die Reporting Collectoren sammeln die Daten und versenden diese an den Reporting-Server. Die Reporting Trigger lösen die Erstellung eines Reports im Reporting-Server aus.

Über das [Runtime-Deployment \[► 44\]](#) werden auch für die Reporting Module, SPS-Bausteine generiert. Dieser SPS-Code kann anschließend in eine TwinCAT Analytics Runtime heruntergeladen werden.

Damit der Reporting-Server auf dem TwinCAT Analytics Runtime-System verfügbar ist, muss das Measurement Setup auf diesem System ausgeführt werden. Nach dem erfolgreichen Setup wird der Reporting-Server als Windows Service gestartet.

Sofern sich die TwinCAT Analytics Runtime auf demselben System befinden, kann der Parameter für die AmsNetId im Konstruktor der SPS-Bausteine leer gelassen werden. Falls dies nicht der Fall ist, muss eine Ads-Verbindung zwischen den zwei Systemen hergestellt werden und die AmsNetId des Systems mit dem Reporting-Server in dem Konstruktor der SPS-Bausteine eingetragen werden.

## 7 Anhang

### 7.1 FAQ – Häufig gestellte Fragen und Antworten

In diesem Abschnitt werden häufig gestellte Fragen beantwortet, um Ihnen die Arbeit mit der TwinCAT Analytics Runtime zu erleichtern. Falls Sie weitere Fragen haben, wenden Sie sich bitte an unser Support-Team [support@beckhoff.com](mailto:support@beckhoff.com).

Ist es möglich, die Anzahl der verbundenen HMI-Clients auf dem Analytics Runtime-Gerät zu erhöhen?

[▶ 170]

**Ist es möglich, die Anzahl der verbundenen HMI-Clients auf dem Analytics Runtime-Gerät zu erhöhen?**

!Ja. Vier HMI-Clients sind in der Analytics Runtime bereits enthalten. Es handelt sich jedoch um ein Standard TwinCAT-System, d. h. Sie können zusätzliche Lizenzen wie TF2030 Client Pack 10 oder TF2040 Client Pack 25 buchen.



Mehr Informationen:  
**[www.beckhoff.com/tf3550](http://www.beckhoff.com/tf3550)**

Beckhoff Automation GmbH & Co. KG  
Hülshorstweg 20  
33415 Verl  
Deutschland  
Telefon: +49 5246 9630  
[info@beckhoff.com](mailto:info@beckhoff.com)  
[www.beckhoff.com](http://www.beckhoff.com)

