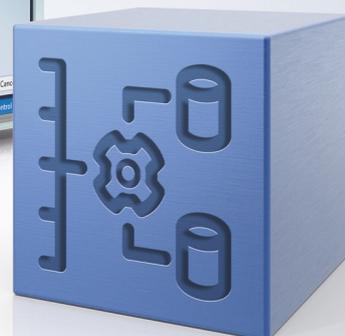
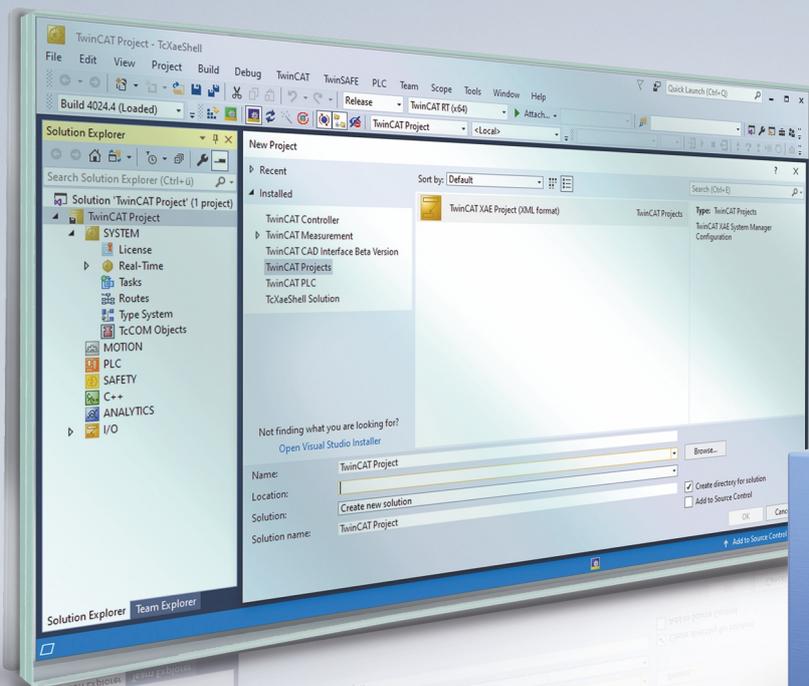


Handbuch | DE

TF3520

TwinCAT 3 | Analytics Storage Provider



Inhaltsverzeichnis

1	Vorwort.....	5
1.1	Hinweise zur Dokumentation	5
1.2	Zu Ihrer Sicherheit.....	6
1.3	Hinweise zur Informationssicherheit	7
2	Übersicht.....	8
3	Installation	10
3.1	Systemvoraussetzungen	10
3.2	Installation	10
3.3	Installation unter TwinCAT4026	13
3.4	Lizenzierung	14
3.5	Installation TwinCAT/BSD	16
4	Analytics Workflow – Erste Schritte.....	17
4.1	Aufzeichnung von Maschinendaten	17
4.2	Kommunikation	20
4.3	Historisierung von Daten	21
4.4	Importierung/Konvertierung von Analytics Files	29
4.5	Analyse der Daten.....	30
4.6	24-stündige Anwendung von Analytics	35
5	Technische Einführung	45
6	Konfiguration.....	47
6.1	Configurator	47
6.1.1	Generic Configurations	48
6.1.2	Additional Configurations	48
6.1.3	Importierung/Konvertierung von Analytics Files.....	57
6.2	Manager	59
6.2.1	Manager („Recorder“)	60
6.2.2	Datenmanagement mit Rule Engine	66
6.3	Arbeit mit historischen Daten	99
6.4	Konsolen- Konfigurator/Client	103
6.4.1	Konfigurator.....	104
6.4.2	Client.....	105
6.4.3	Batch Files zur Steuerung.....	108
7	SPS-API.....	114
7.1	Funktionsbausteine	114
7.1.1	Topic-Architektur	114
7.1.2	FB_ALY_StorageProvider.....	135
7.2	Datentypen	145
7.2.1	ST_ALY_SP_Config.....	145
7.2.2	ST_ConnectionSettings	146
7.2.3	E_CancelType.....	146
7.2.4	E_ControlMode	147
7.2.5	E_DataStorageType.....	147
7.2.6	E_RawDataFormat.....	148

7.2.7	E_RecordingState	148
7.2.8	E_RecordMode	148
7.2.9	E_ReloadType	149
7.2.10	E_RingBufferMode	149
7.2.11	E_SetGetHistoricalDataState	149
7.2.12	E_StorageState	150
7.2.13	E_SymbolMode	150
7.2.14	E_PipelineCmdType	151
8	Beispiele	152
8.1	SPS-Client	152
9	Anhang	160
9.1	Glossar	160
9.2	FAQ – Häufig gestellte Fragen und Antworten	162
9.3	Third-party components	163
10	Support und Service	164

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®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® und XTS® 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 Kennzeichnungen führen.



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.

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

Fremdmarken

In dieser Dokumentation können Marken Dritter verwendet werden. Die zugehörigen Markenvermerke finden Sie unter: <https://www.beckhoff.com/trademarks>.

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

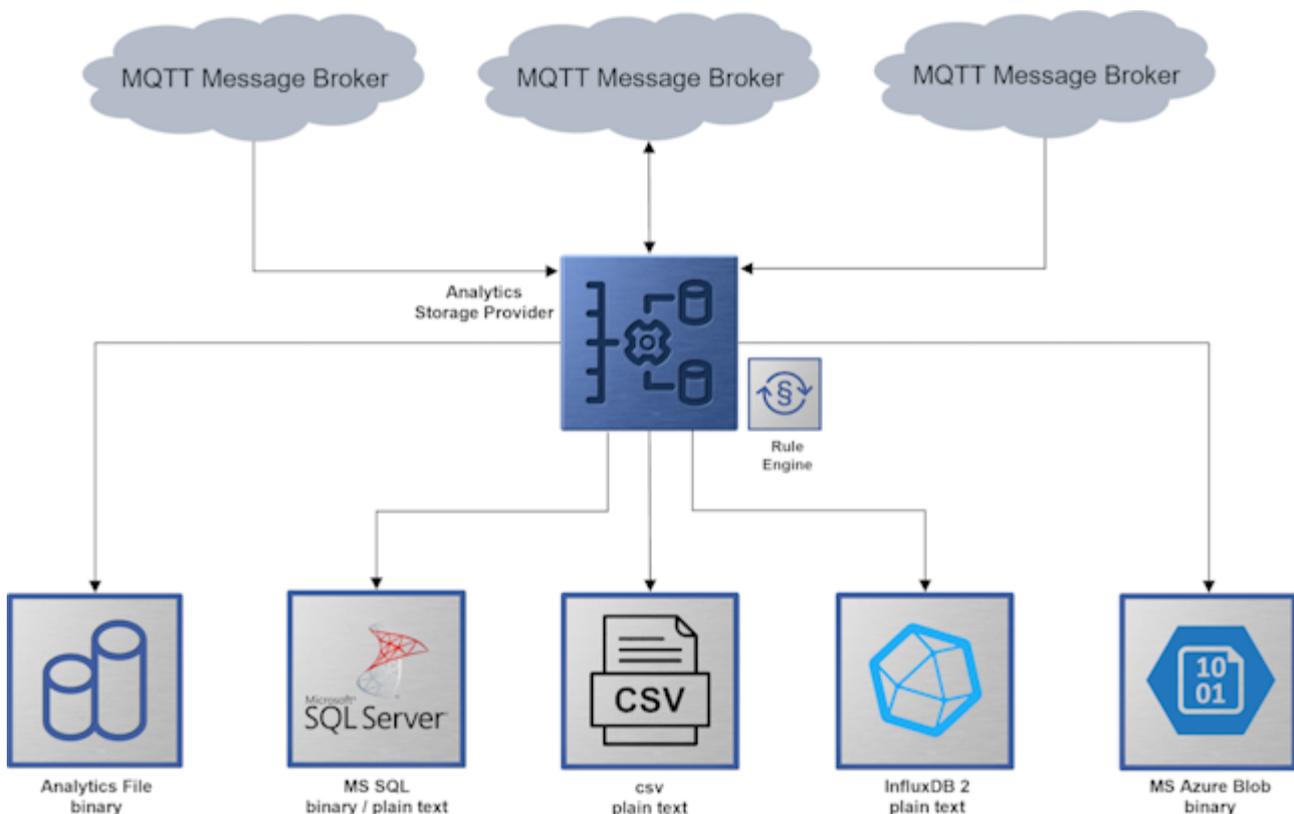
Beckhoff bietet mit dem TwinCAT Analytics Storage Provider eine Möglichkeit, hochaufgelöste Daten in einem Binärformat, oder im Klartext, kontinuierlich abzulegen. Entscheidend ist, dass sich der Anwender keine Gedanken zur Datenablage machen muss. Darum kümmert sich der Storage Provider automatisch. Die Konfiguration erfolgt mit wenigen Klicks im Engineering. Aufwendige SQL-Befehle sind nicht notwendig. Es können klassische Datenbanken, aber auch binäre Blob-Stores verwendet werden.

Komponenten

- TwinCAT Analytics Storage Provider Service: Ein Windows Service, der die Kommunikation verwaltet.
- TwinCAT Analytics Storage Provider PLC Library: Eine TwinCAT 3 SPS-Bibliothek mit Funktionen zur Steuerung des Storage Providers aus einer SPS-Anwendung.
- TwinCAT Analytics Storage Provider Client: Eine Konsolen Anwendung mit Funktionen zur Steuerung des Storage Providers über die Kommandozeile.
- TwinCAT Analytics Storage Provider Configurator: Eine Engineering Anwendung zum Konfigurieren der verschiedenen Datenbanken und Datenquellen.
- TwinCAT Analytics Storage Provider Manager: Eine Engineering Anwendung zum Verwalten der aufgezeichneten Daten und zur Steuerung des Storage Providers.

Funktionsprinzip

Der Analytics Storage Provider empfängt und sendet Daten via MQTT Kommunikationsprotokoll. Dazu ist er mit einem nativen MQTT-Message-Broker im Netzwerk verbunden und auf der anderen Seite mit der entsprechenden Datenbank.



Unterstützte Datenbanken/Storage

- [TwinCAT Analytics Binary File](#) [► 50]
- [Microsoft SQL \(Binär-Format / Klartext\)](#) [► 51]
- [CSV-Datei](#) [► 53]
- [InfluxDB](#) [► 54]

- [Microsoft Azure Blob](#) [▶_55]

3 Installation

3.1 Systemvoraussetzungen

Die Anforderungen des Service und der PLC Library des Analytics Storage Providers finden Sie in den folgenden Tabellen. Beide können auch auf einem System installiert werden.

Technische Daten Service	TF3520 TwinCAT 3 Analytics Storage Provider
Zielsystem	Windows 10, TwinCAT/BSD
.NET Framework	.Net 4.5.1 oder höher
Min. TwinCAT-Version	3.1.4022.25
Min. TwinCAT-Level	TC1000 TwinCAT 3 ADS

Technische Daten Library	TF3520 TwinCAT 3 Analytics Storage Provider
Zielsystem	Windows 10, TwinCAT/BSD
Min. TwinCAT-Version	3.1.4022.29
Min. TwinCAT-Level	TC1200 TwinCAT 3 PLC

3.2 Installation

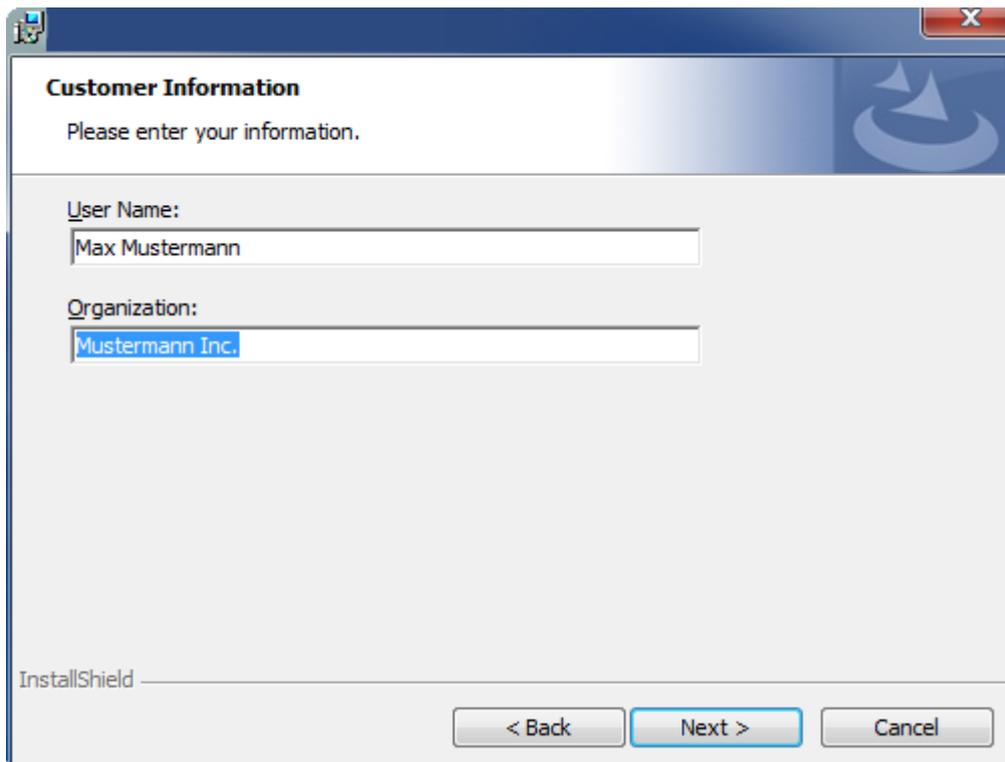
Setup-Installation (TwinCAT 3.1 Build 4024)

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.

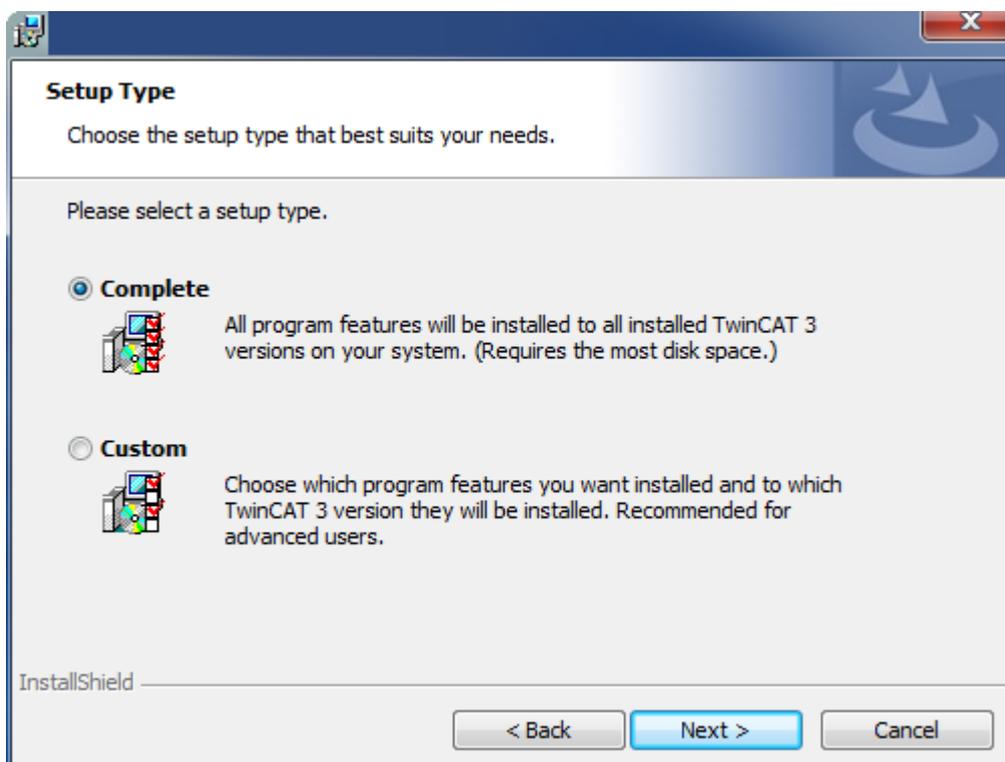
User Name:
Max Mustermann

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.

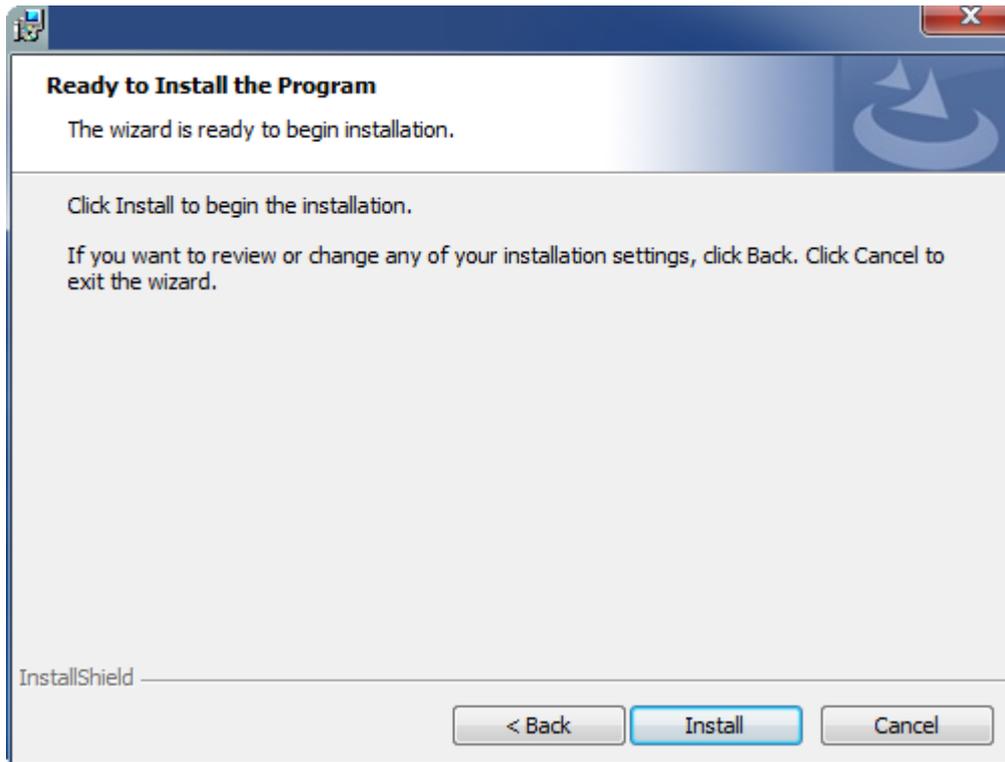
Complete
 All program features will be installed to all installed TwinCAT 3 versions on your system. (Requires the most disk space.)

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

InstallShield

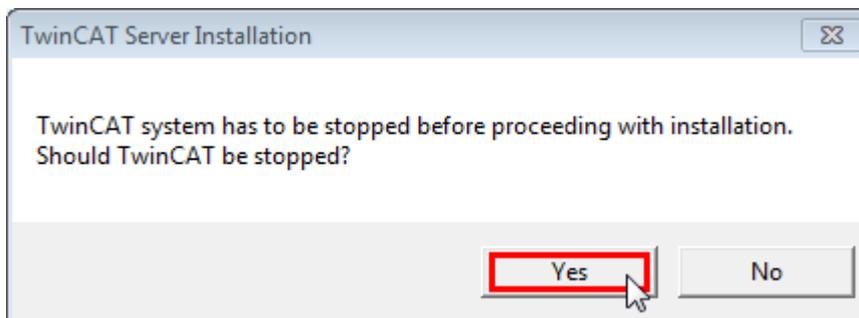
< Back Next > Cancel

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

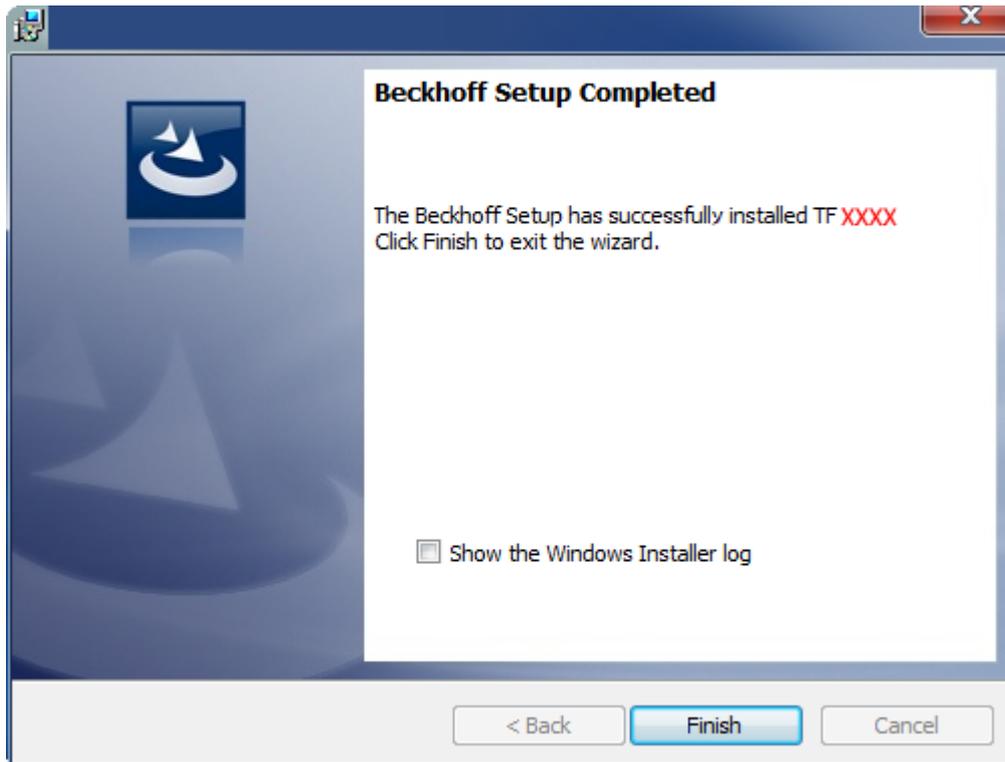


⇒ Ein Dialog weist Sie darauf hin, dass das TwinCAT-System für die weitere Installation gestoppt werden muss.

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 Installation unter TwinCAT4026

TwinCAT Package Manager

Wenn Sie TwinCAT 3.1 Build 4026 (und höher) auf dem Betriebssystem Microsoft Windows verwenden, können Sie diese Function über den TwinCAT Package Manager installieren, siehe [Dokumentation zur Installation](#).

Normalerweise installieren Sie die Function über den entsprechenden Workload; dennoch können Sie die im Workload enthaltenen Pakete auch einzeln installieren. Diese Dokumentation beschreibt im Folgenden kurz den Installationsvorgang über den Workload.

Kommandozeilenprogramm TcPkg

Über das TcPkg **Command Line Interface (CLI)** können Sie sich die verfügbaren Workloads auf dem System anzeigen lassen:

```
tcpkg list TF3520
```

Über das folgende Kommando können Sie den Workload der TF3520 TC3 Analytics Storage Provider-Function installieren.

```
tcpkg install TF3520.AnalyticsStorageProvider.XAE  
tcpkg install TF3520.AnalyticsStorageProvider.XAR
```

TwinCAT Package Manager UI

Über das **User Interface (UI)** können Sie sich alle verfügbaren Workloads anzeigen lassen und diese bei Bedarf installieren.

Folgen Sie hierzu den entsprechenden Anweisungen in der Oberfläche.

3.4 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

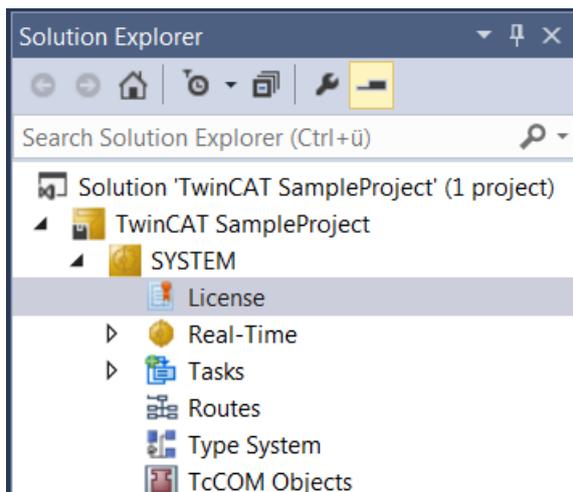
Die Beschreibung der Lizenzierung einer Vollversion finden Sie im Beckhoff Information System in der Dokumentation „[TwinCAT-3-Lizenzierung](#)“.

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.

- 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“).

Order No	License	Add License
TF3601	TC3 Condition Monitoring Level 2	<input type="checkbox"/> cpu license
TF3650	TC3 Power Monitoring	<input type="checkbox"/> cpu license
TF3680	TC3 Filter	<input type="checkbox"/> cpu license
TF3800	TC3 Machine Learning Inference Engine	<input type="checkbox"/> cpu license
TF3810	TC3 Neural Network Inference Engine	<input type="checkbox"/> cpu license
TF3900	TC3 Solar-Position-Algorithm	<input type="checkbox"/> cpu license
TF4100	TC3 Controller Toolbox	<input checked="" type="checkbox"/> cpu license
TF4110	TC3 Temperature-Controller	<input type="checkbox"/> cpu license
TF4500	TC3 Speech	<input type="checkbox"/> cpu license

- Öffnen Sie die Registerkarte **Order Information (Runtime)**.
 ⇒ In der tabellarischen Übersicht der Lizenzen wird die zuvor ausgewählte Lizenz mit dem Status „missing“ angezeigt.
- Klicken Sie auf **7 Days Trial License...**, um die 7-Tage-Testlizenz zu aktivieren.

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

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

3.5 Installation TwinCAT/BSD

Der TwinCAT 3 Analytics Storage Provider Server steht als Package für TwinCAT/BSD im Package-Repository bereit. Unter dem Package Namen „TF3520-Analytics-Storage-Provider“ kann er über folgenden Befehl installiert werden:

```
doas pkg install TF3520-Analytics-Storage-Provider
```

Weitere Informationen über den [Package Server](#) finden Sie im Bereich Embedded-PC im TwinCAT/BSD-Handbuch.

Nach einem Neustart des Systems oder Restart von TwinCAT, wird der TwinCAT 3 Analytics Storage Provider Server ebenfalls gestartet und ist über MQTT von einem Client konfigurierbar.

● MQTT-Port-Freigabe



Für die Verwendung des Analytics Storage Providers und des [Konsolen- Konfigurator/Client](#) [► 103] unter TwinCAT/BSD muss der entsprechende MQTT-Port für die Kommunikation freigegeben werden. Weitere Infos siehe: [Portfreigabe unter TwinCAT/BSD](#)

Nach der Installation befindet sich die Client.dll für die Konsole unter dem Pfad `/usr/local/etc/TwinCAT/Functions/TF3520-Analytics-Storage-Provider/Client`.

Der Analytics Storage Provider Service kann bei aktivierter Lizenz durch folgenden Befehl gestartet werden:

```
doas service TcAnalyticsStorageProvider start
```

4 Analytics Workflow – Erste Schritte

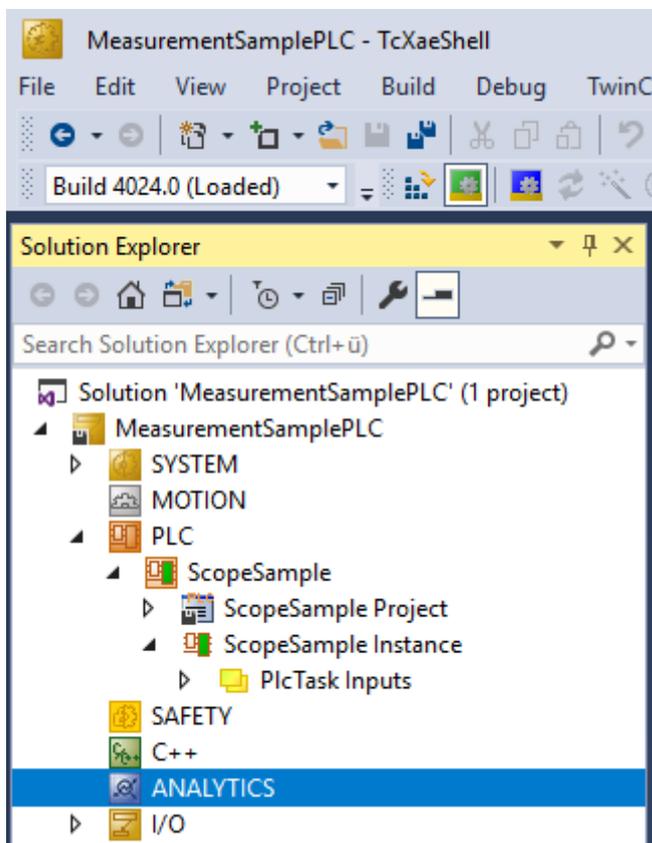
Diese Dokumentation präsentiert Schritt für Schritt den vollständigen TwinCAT Analytics Workflow. Von der Datenerfassung über die Kommunikation und Historisierung bis hin zur Auswertung und Analyse der Daten sowie zur Präsentation der Daten im webbasierten Dashboard.

4.1 Aufzeichnung von Maschinendaten

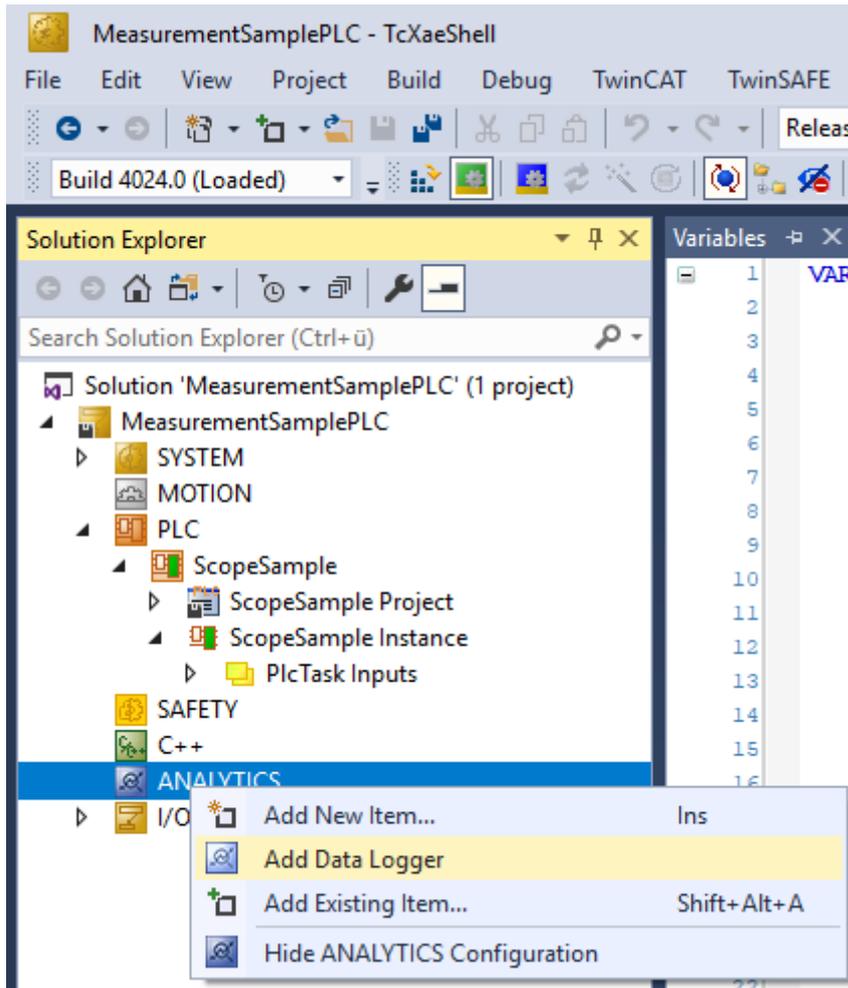
Maschinenseitig ist der Analytics Logger die Aufzeichnungseinrichtung von Prozessdaten des Maschinenabbaus, SPS, NC usw. Der Logger arbeitet im Echtzeitkontext von TwinCAT.

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.

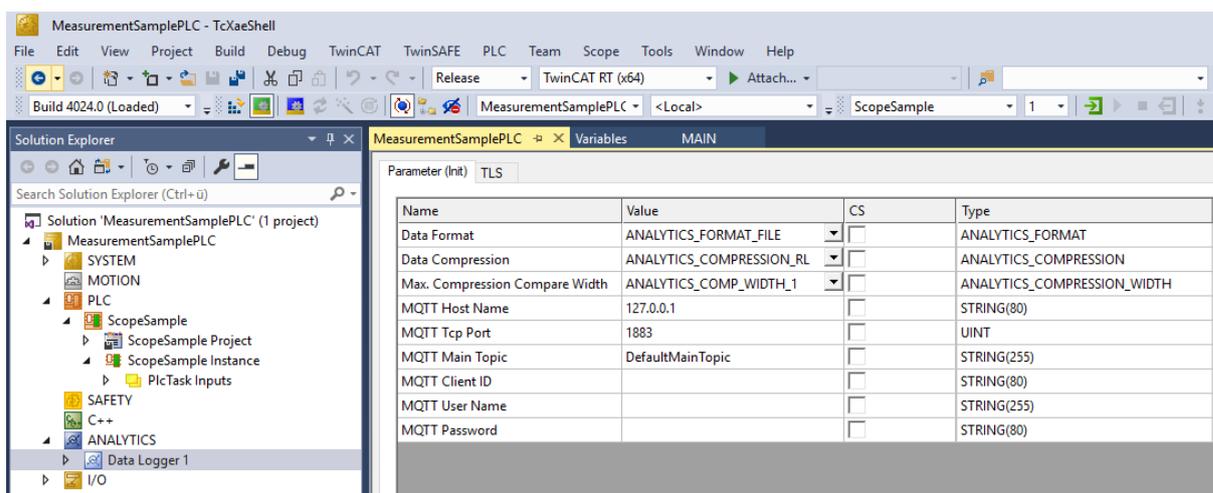
- Um den Analytics Logger zu konfigurieren, müssen Sie in Ihrem vorhandenen TwinCAT-Projekt zum Analytics-Baumknoten navigieren.



- 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.

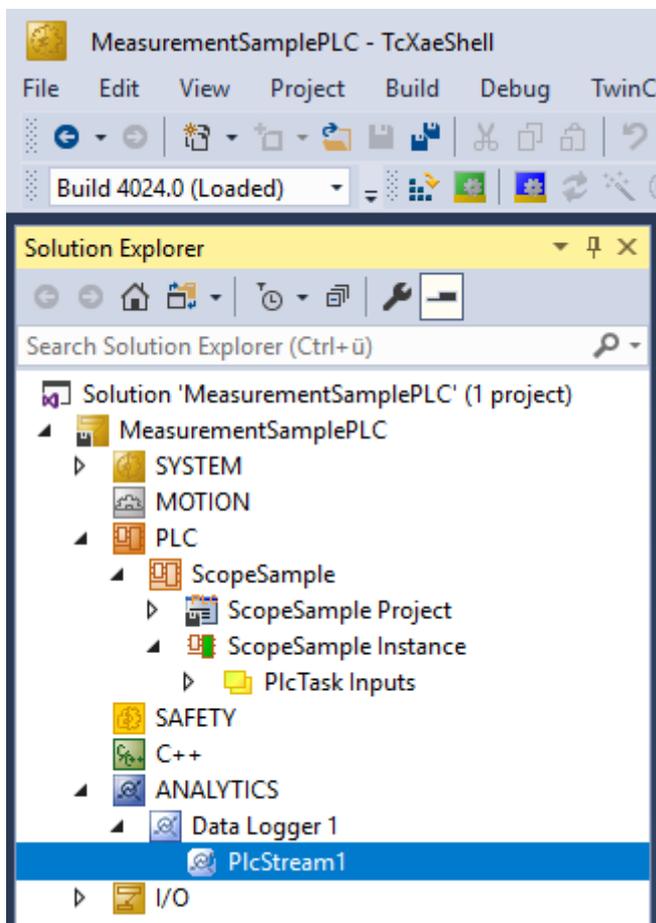
-FILE-Format: Der Analytics Logger speichert die Daten in lokalen Binärdateien und alle anderen Einstellungen sind nicht mehr notwendig. Die Dateien werden in `C:\TwinCAT\3.1\Boot\Analytics` gespeichert.

-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.

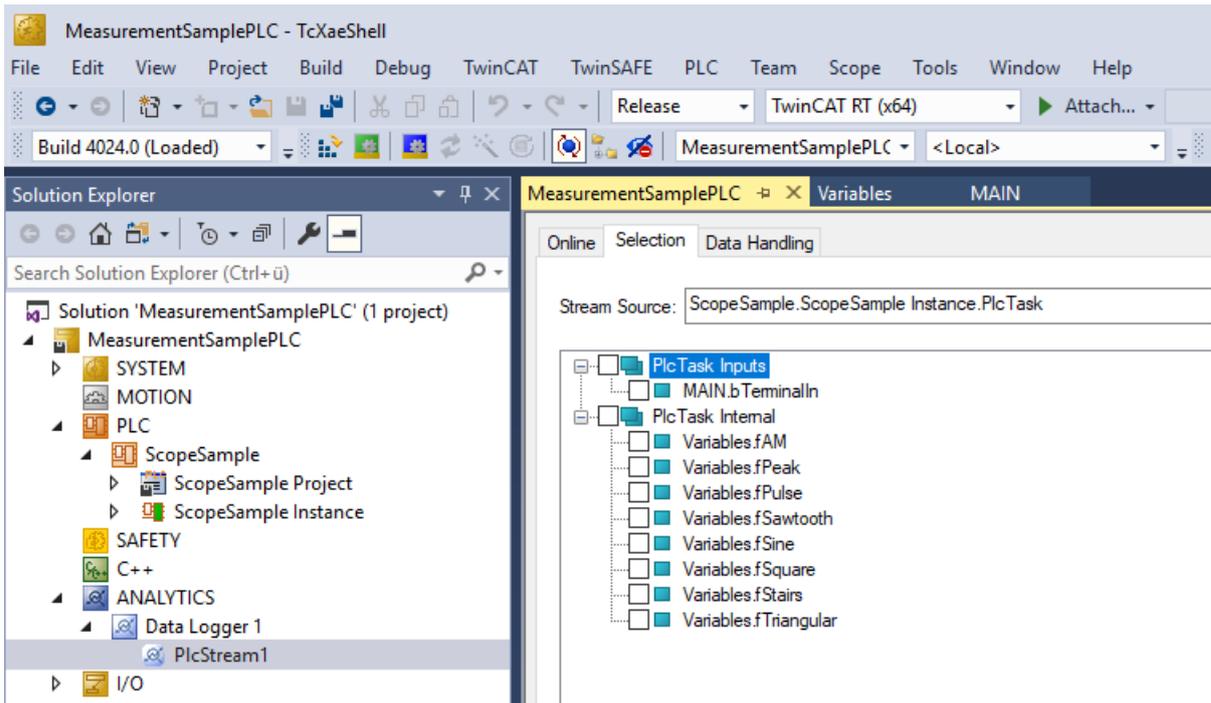
-Auf der Registerkarte **TLS** (Transport Layer Security) können die Sicherheitseinstellungen konfiguriert werden. TLS ist ein sicherer Kommunikationskanal zwischen Client und Server. Bei Verwendung von Zertifikaten ist der TCP-Port 8883 ausschließlich für MQTT über TLS reserviert! Der Analytics Logger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

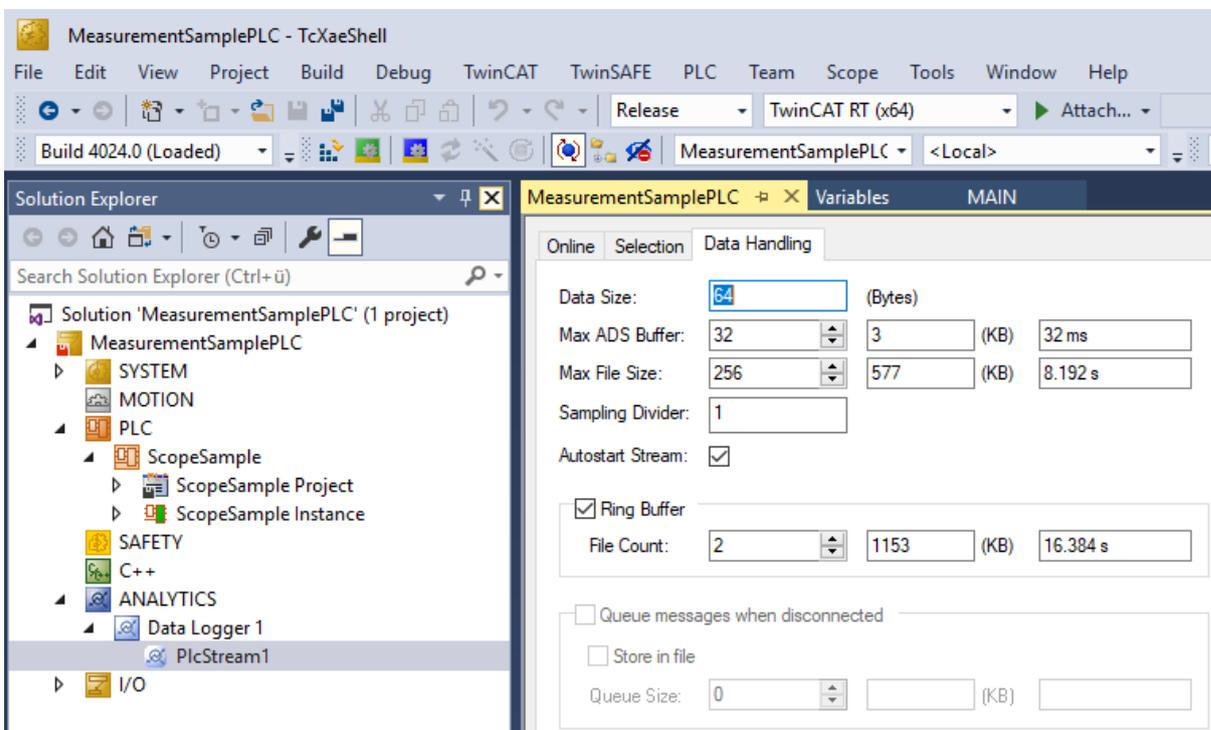


Ein zusätzlicher Geräte-Stream wird angezeigt, wenn Ihre Konfiguration ein EtherCAT-Prozessabbild liefert.

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

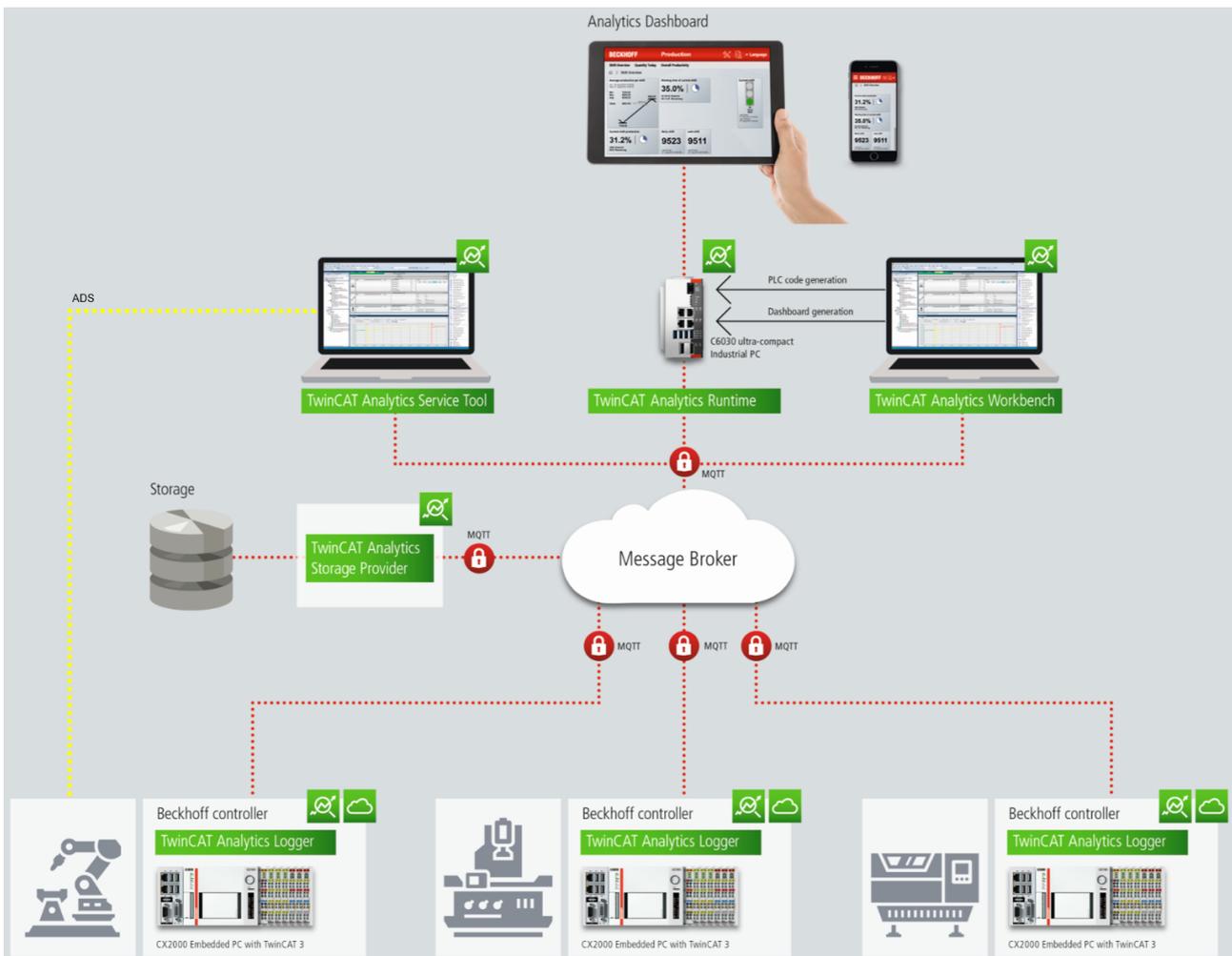


- Schließlich ist es im Karteireiter **Data Handling** möglich, die Paketgröße für die Frames zu ändern oder den Ringpuffer für Verbindungsunterbrechungen und Datei zu konfigurieren.



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`.

The screenshot shows the 'TwinCAT Analytics Storage Provider Configuration' dialog box. It is organized into several sections:

- Analytics Storage Provider:** Contains two text input fields. 'MainTopic' is set to 'Beckhoff/' and 'Comment' is set to 'TwinCAT Analytics StorageProvider ("MyDevice")'.
- Messagebroker:** Features a 'Settings' button and the text 'Set connection settings for message broker'.
- Storage:** Includes a 'Type' dropdown menu currently set to 'AnalyticsFile'. Below it is a 'Connectionstring' text area containing '\\beckhoff.com\dfs\UserHomeDir\PascalD\Storage'. To the right of this area is an 'Analytics Folder' button. At the bottom right of this section is an 'Additional Properties' button.
- Logging:** Contains two checkboxes: 'Trace to EventLog' (unchecked) and 'Additional Debug Log' (checked).
- Analytics Storage Provider Windows Service:** Has a dropdown menu set to 'Automatic', 'Start' and 'Stop' buttons, and the status 'Stopped'.

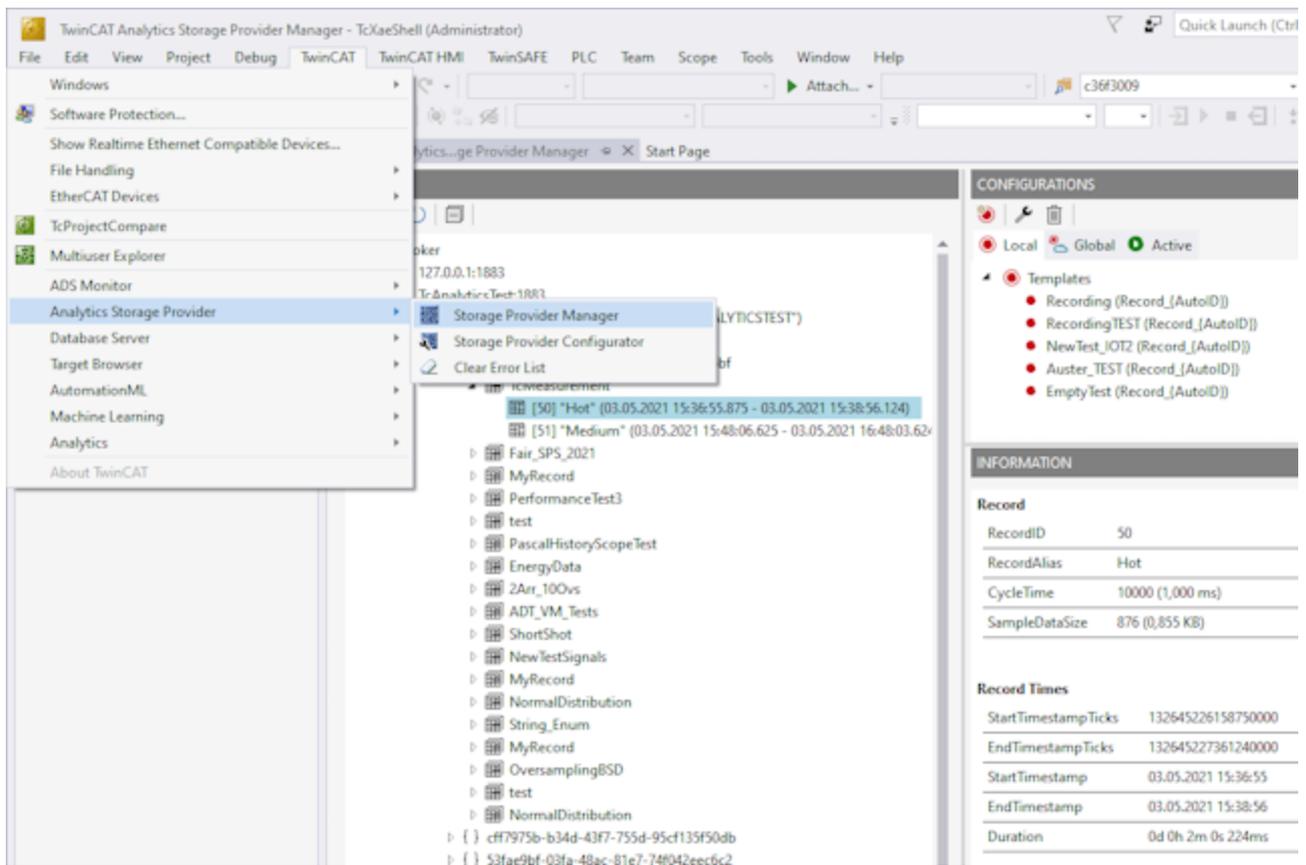
At the bottom of the dialog are three buttons: 'Cancel', 'Save Config', and 'OK'.

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.

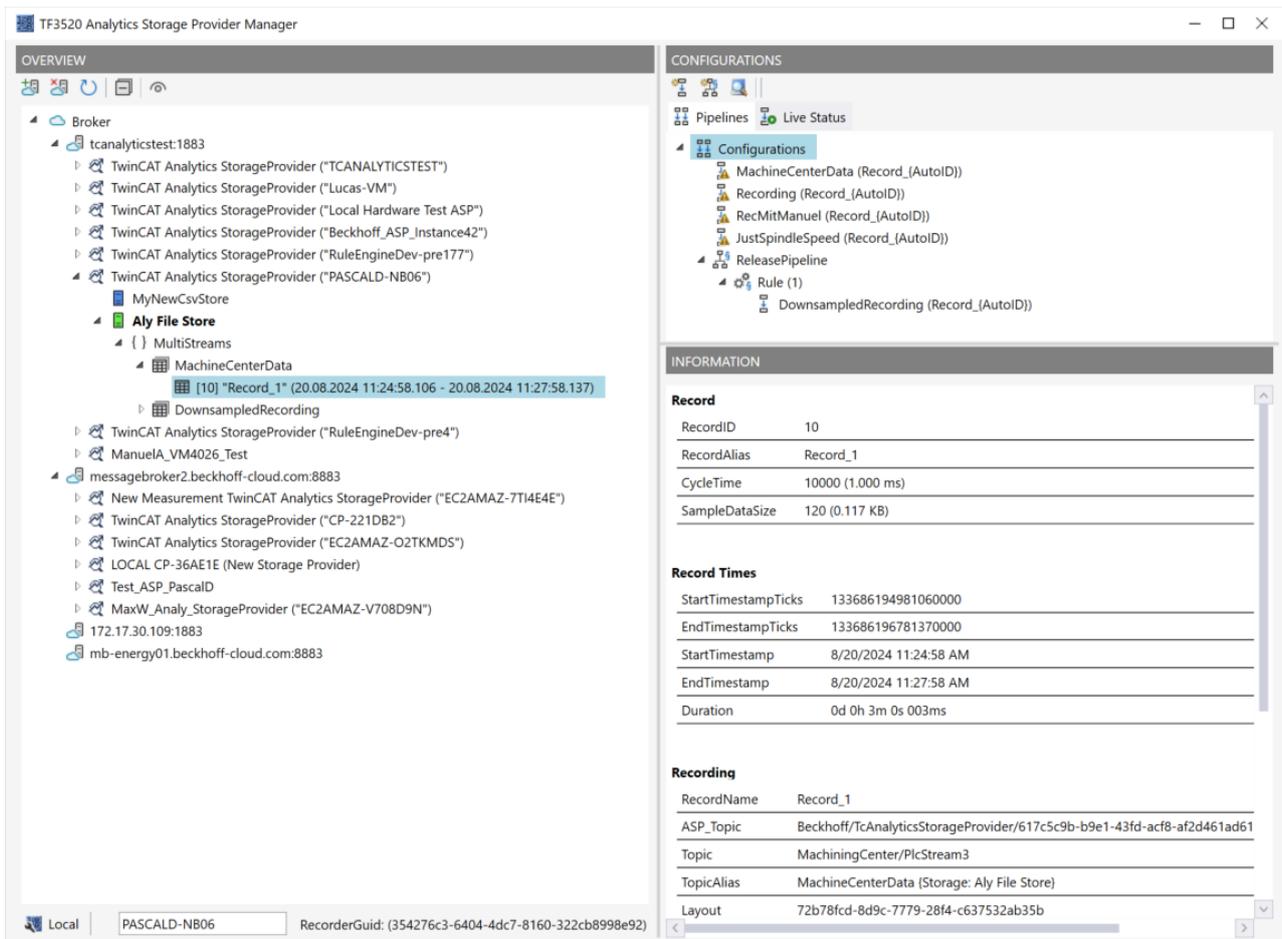
Sie können die Message-Broker-Einstellungen vornehmen und sich für einen Speichertyp entscheiden:

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

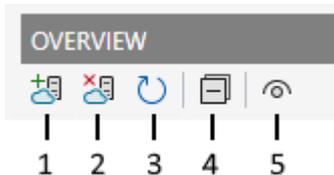
Zuletzt können Sie die Konfiguration speichern und den Dienst starten. Der nächste Schritt besteht in der Konfiguration der spezifischen Aufzeichnung. Hierzu sollten Sie in Ihrer Entwicklungsumgebung den **Storage Provider Manager** auswählen.



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 Provider und historisierten Daten werden übersichtlich dargestellt.



Symbolleiste Manager Fenster („OVERVIEW“)

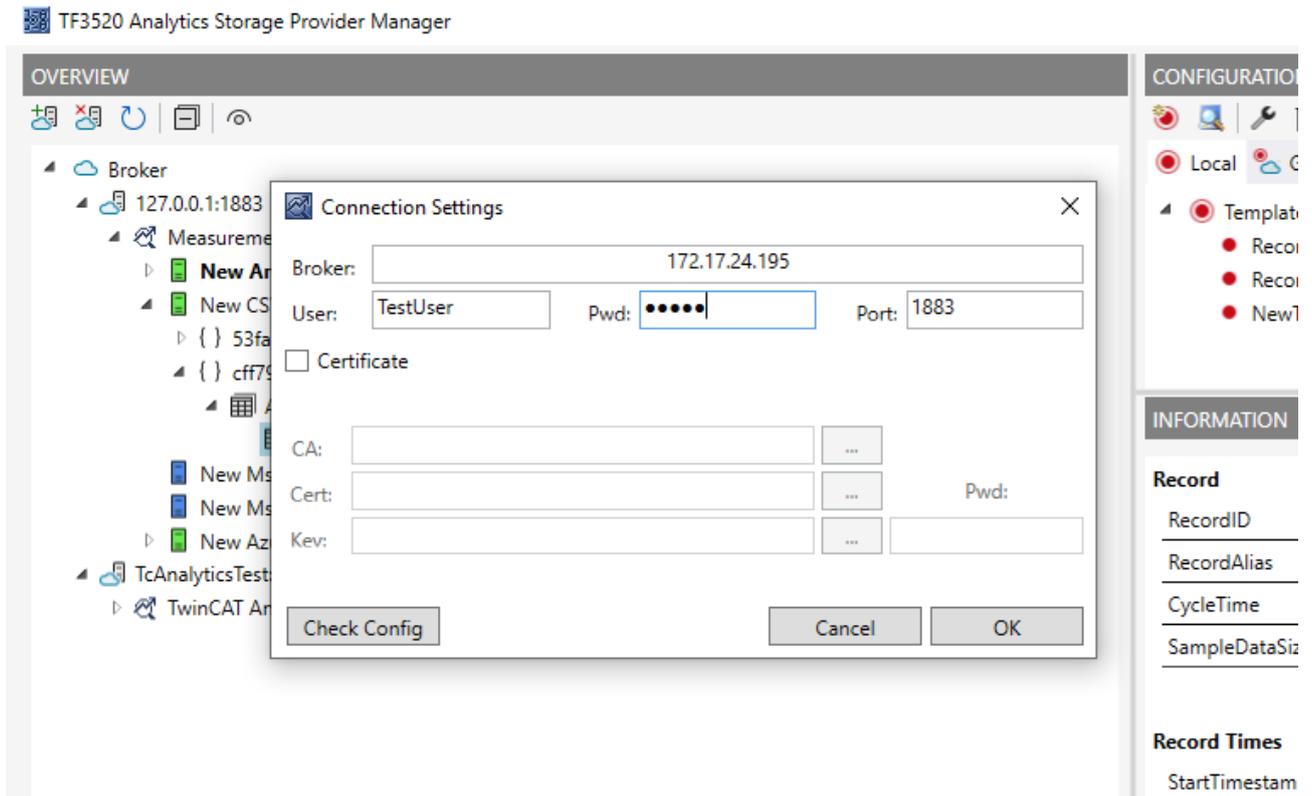


1	Neuen Broker hinzufügen
2	Ausgewählten Broker entfernen
3	Anzeige aktualisieren
4	Alle Knoten einklappen
5	Ansicht Wechsel zwischen Dark/Light Modus

Funktion Manager Fenster („OVERVIEW“)

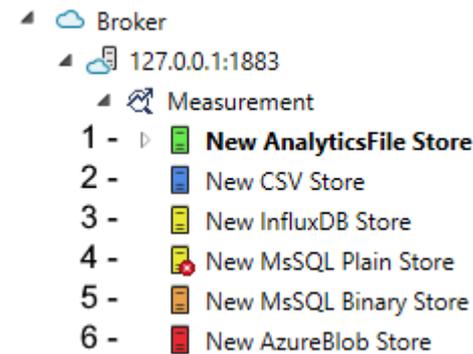
Vergeben Sie zunächst einen „RecorderAlias“. Dieser hilft die gestarteten Aufnahmen zu gruppieren und seine selbst gestarteten wieder zu finden.

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



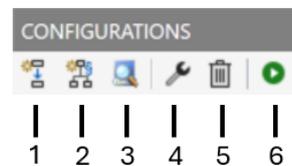
Sobald eine Verbindung zum Broker hergestellt werden konnte, werden alle darauf verbundenen Analytics Storage Provider aufgelistet.

„Storage“ Status



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

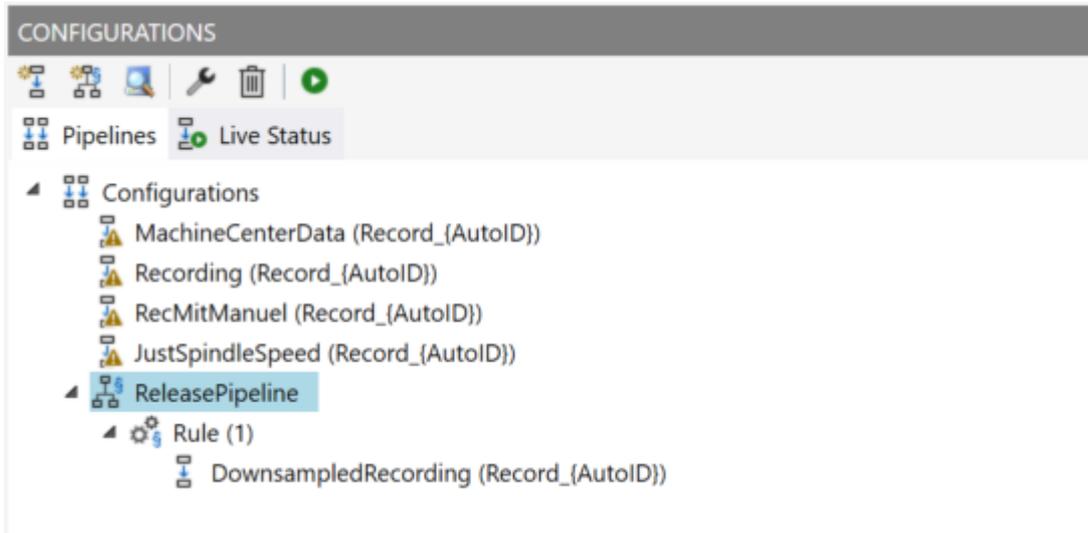
Symbolleiste Manager Fenster („CONFIGURATIONS“)



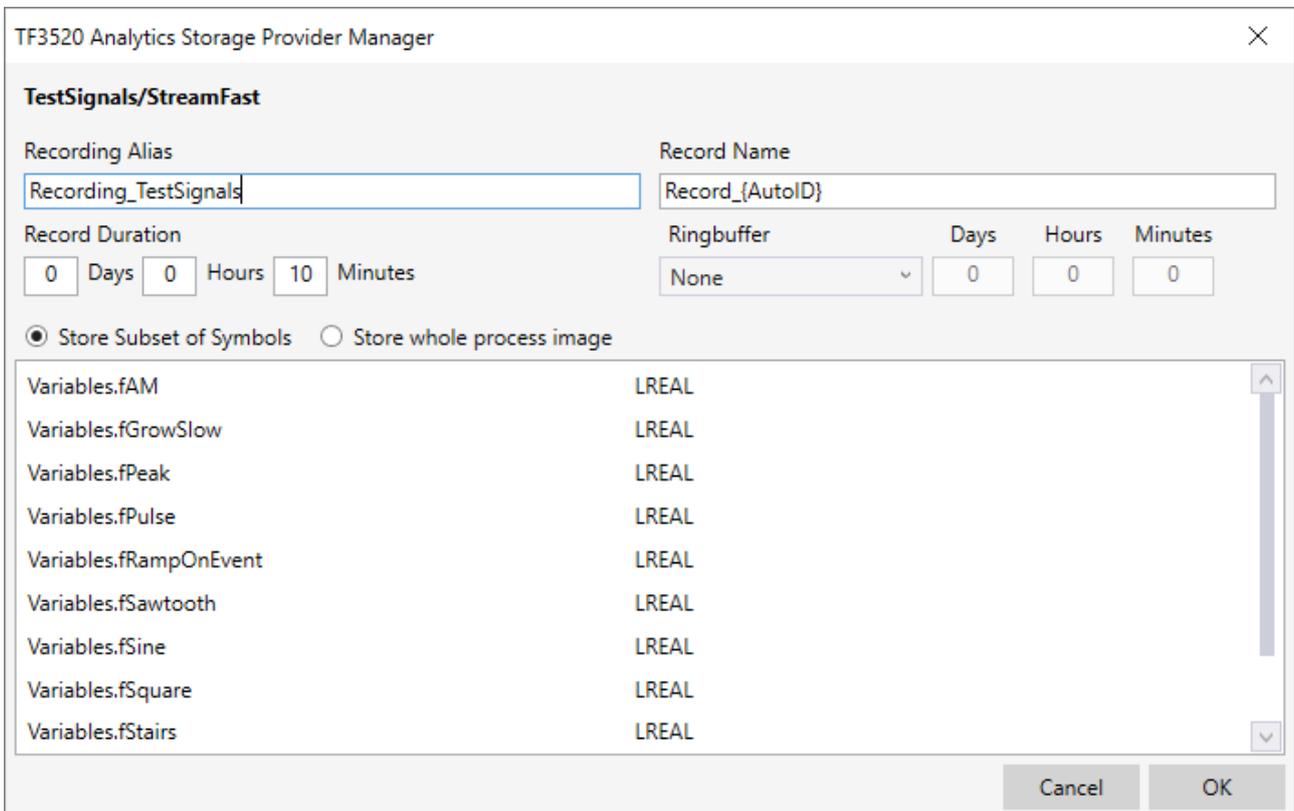
1	Eine neue Pipeline erzeugen
2	Eine neue Pipeline mit Rule Engine erzeugen
3	Target Browser öffnen für Beschaltung einfacher Pipelines
4	Eine ausgewählte Pipeline editieren
5	Eine ausgewählte Pipeline löschen
6	Eine ausgewählte Pipeline starten

Funktion Manager Fenster („CONFIGURATIONS“)

Das Fenster teilt sich in zwei Tabs auf. Pipelines und Live Status. Unter Pipelines finden Sie die Konfigurationen Ihrer Pipelines. Von hier können Sie neue Pipelines definieren, bestehende editieren, löschen oder starten.



Um eine neue einfache Pipeline zu erzeugen, drücken Sie den Button **Neue Pipeline erzeugen**. Folgender Dialog öffnet sich.



Sie können nun die Symbole, die aufgezeichnet werden sollen, vom Target Browser per Drag-and-Drop in den Dialog ziehen. Vergeben Sie zusätzlich ein Recording Alias und einen Record Name.

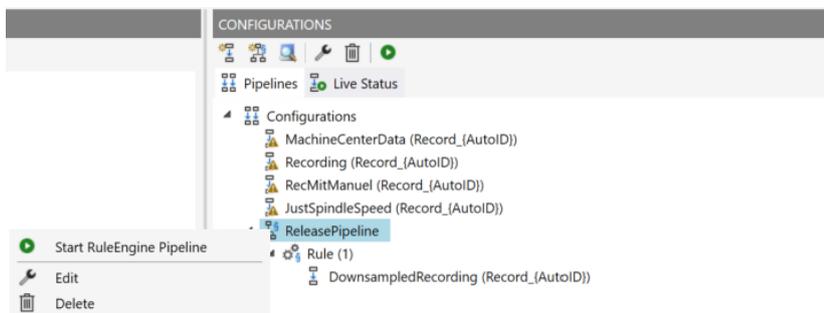
Für den Record Name stehen verschiedene Platzhalter zur Verfügung:

"{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 Speicherplatz oder Zeit eingestellt werden.

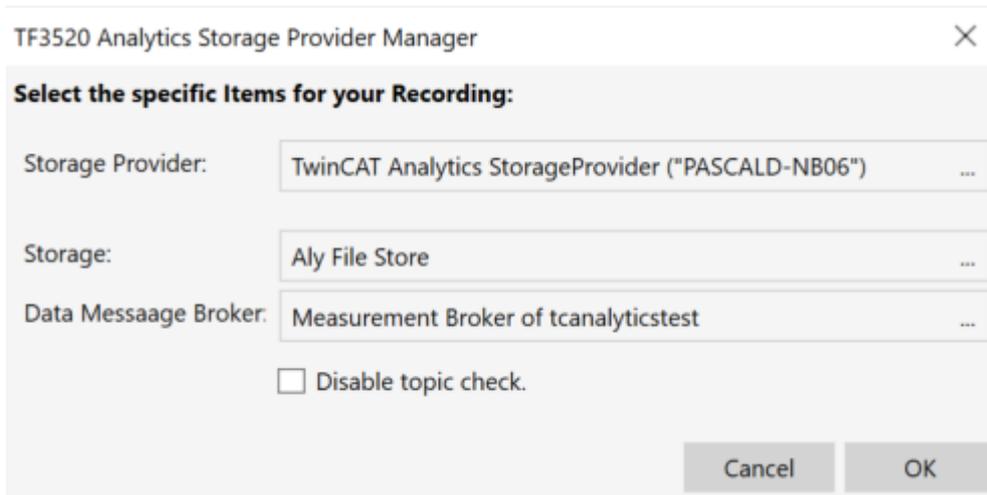
Mit **OK** werden die Eingaben bestätigt und eine neue lokale Aufnahmedefinition wird erzeugt.

Es ist nun möglich diese Definition direkt über die Toolbar oder das Kontextmenü zu starten.



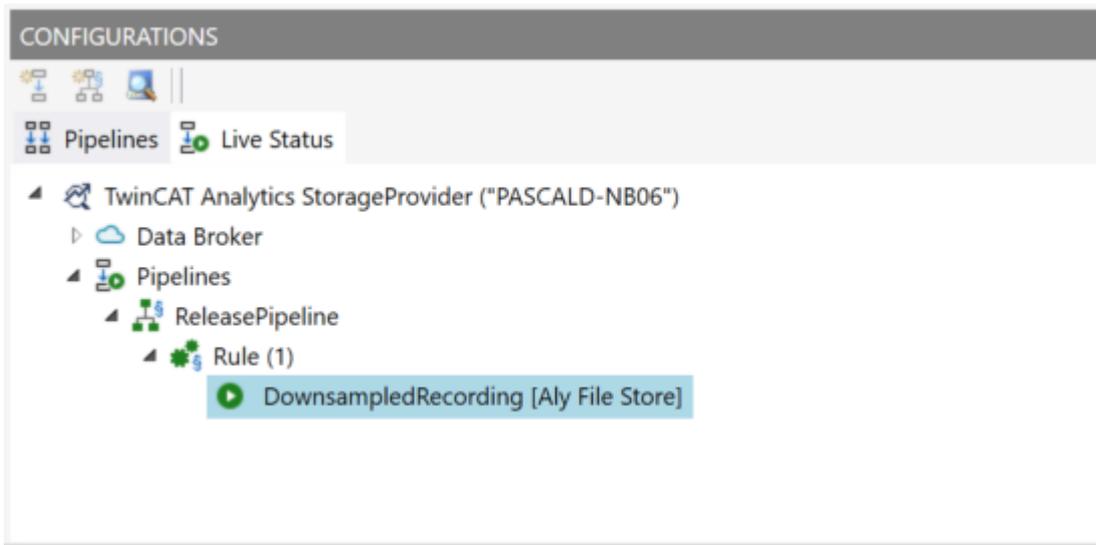
Es besteht aber auch die Möglichkeit die Definition global zugänglich zu machen. Dies kann über das Kontextmenü mit dem Eintrag **Publish Recording** gemacht werden.

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 Data 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, gefunden werden.

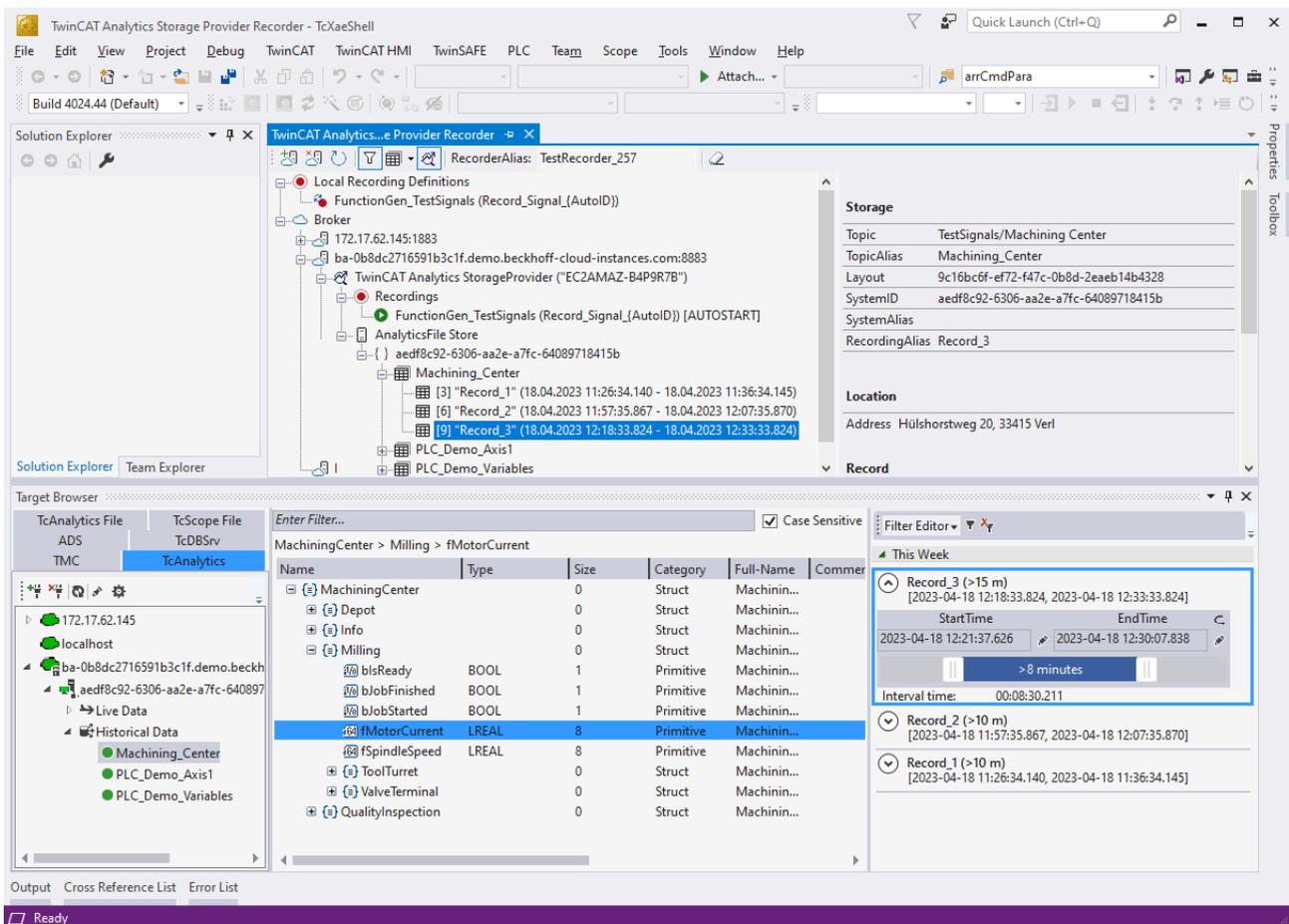
Nach dem Start einer Pipeline springt die Ansicht automatisch in den zweiten Tab, den Live Status.



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 Importierung/Konvertierung von Analytics Files

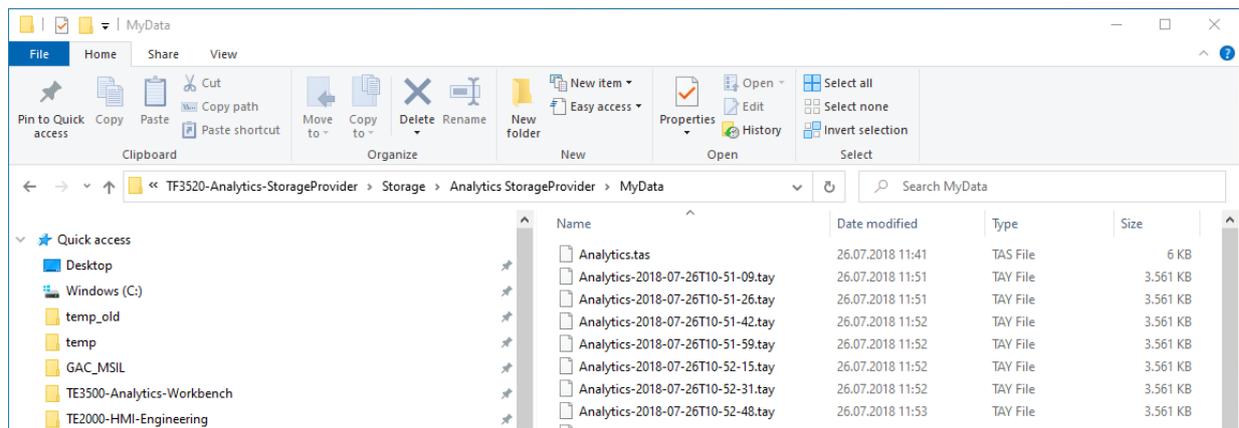


Im Folgenden wird davon ausgegangen, dass Sie TwinCAT unter „C:/TwinCAT“ installiert haben. Andernfalls müssen Sie die angegebenen Pfade entsprechend anpassen.

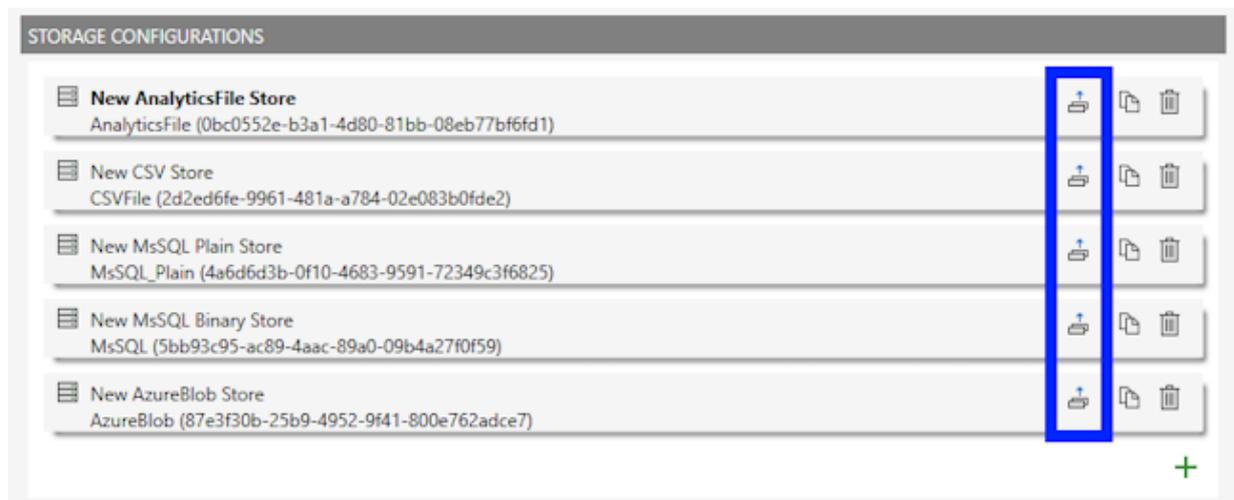
Sie können Aufnahmen aus dem Analytics Logger, die im Analytics File Format (*Analytics.tas*, *Analytics-<Date>.tay*) gespeichert sind, in den Storage Provider importieren. Generell können Sie Daten, die auch vom Storage Provider als Analytics File abgespeichert wurden, in andere Formate konvertieren. Analytics File ist dabei immer das Quellformat.

Führen Sie dazu folgende Schritte durch:

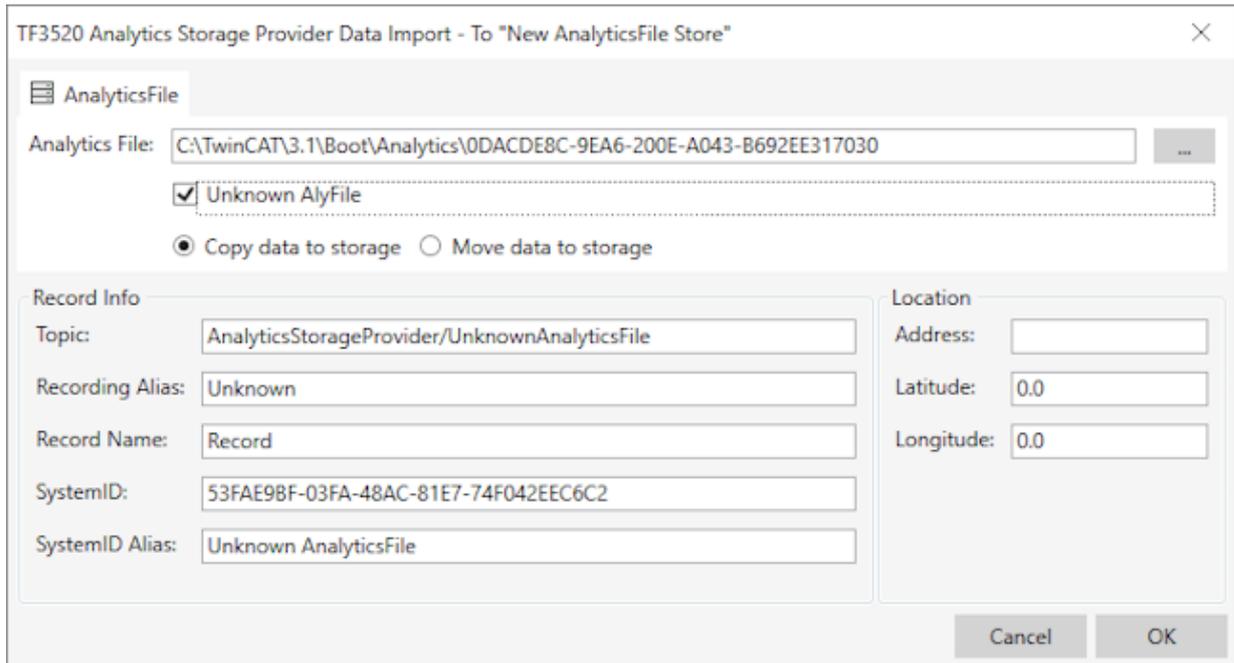
1. Legen Sie den Ordner mit den Analytics Dateien in Ihrem Storage Provider Speicherort ab, Standardmäßig hier: *C:\TwinCAT\Functions\TF3520-Analytics-StorageProvider\Storage\Analytics StorageProvider* (legen Sie die Ordner manuell an, falls diese nicht vorhanden sind)



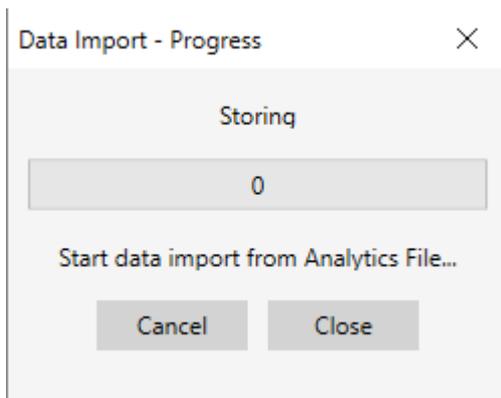
2. Öffnen Sie die *TwinCAT.Analytics.StorageProvider.Configurator.exe*. Das Programm finden Sie unter dem Pfad *C:\TwinCAT\Functions\TF3520-Analytics-StorageProvider\Tools*
3. Wählen Sie den Storage aus, in den das Analytics File importiert werden soll und drücken Sie den Button **DataImport**.



4. Wählen Sie dann im DataImport Dialog den Pfad zum Analytics File aus und geben alle weiteren bekannten Parameter zu dieser Aufnahme an.



5. Drücken Sie **OK** und der Datenimport beginnt.

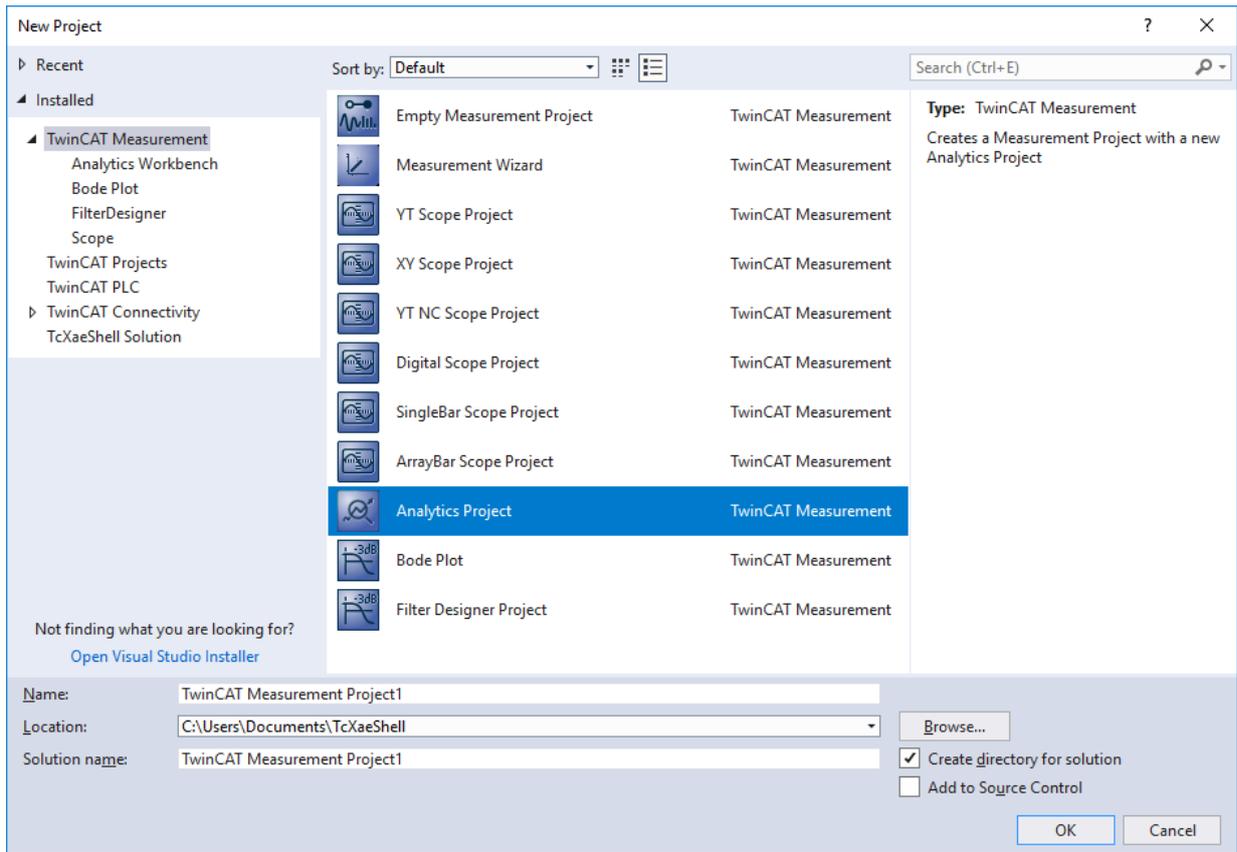


- ⇒ Nun können Sie Ihre importierten Daten im [TwinCAT Target Browser \[► 99\]](#) sehen.
- ⇒ Möglicherweise müssen Sie kurz warten bzw. Ihren Storage Provider einmal neu starten.

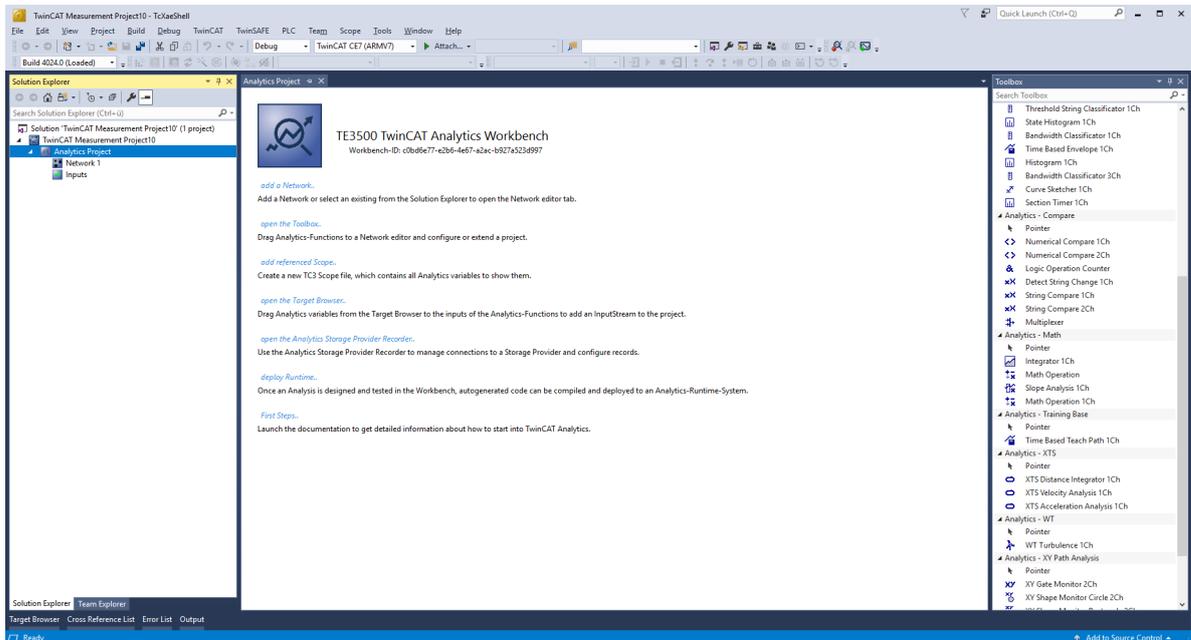
4.5 Analyse der Daten

- ✓ Öffnen Sie Ihre TwinCAT Engineering-Umgebung, um die Analyse der Daten zu starten.
1. Öffnen Sie **Visual Studio® > File > New > Project...**

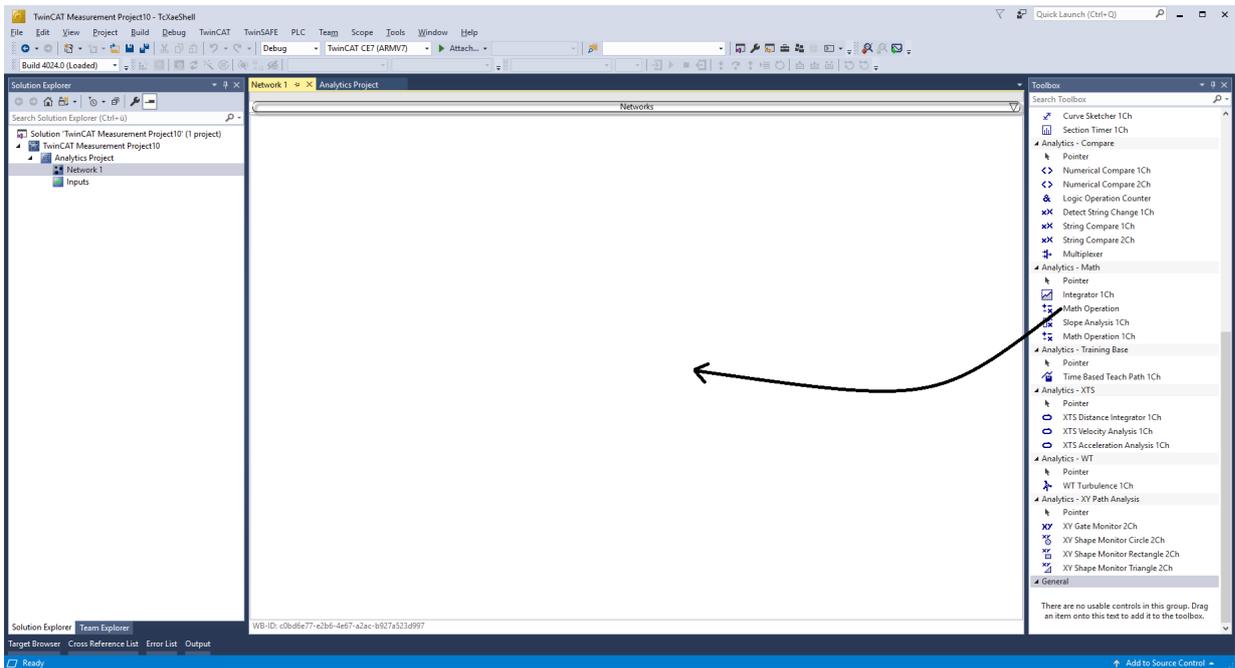
2. Wählen Sie aus **TwinCAT Measurement** die **Analytics-Projektvorlage** aus.



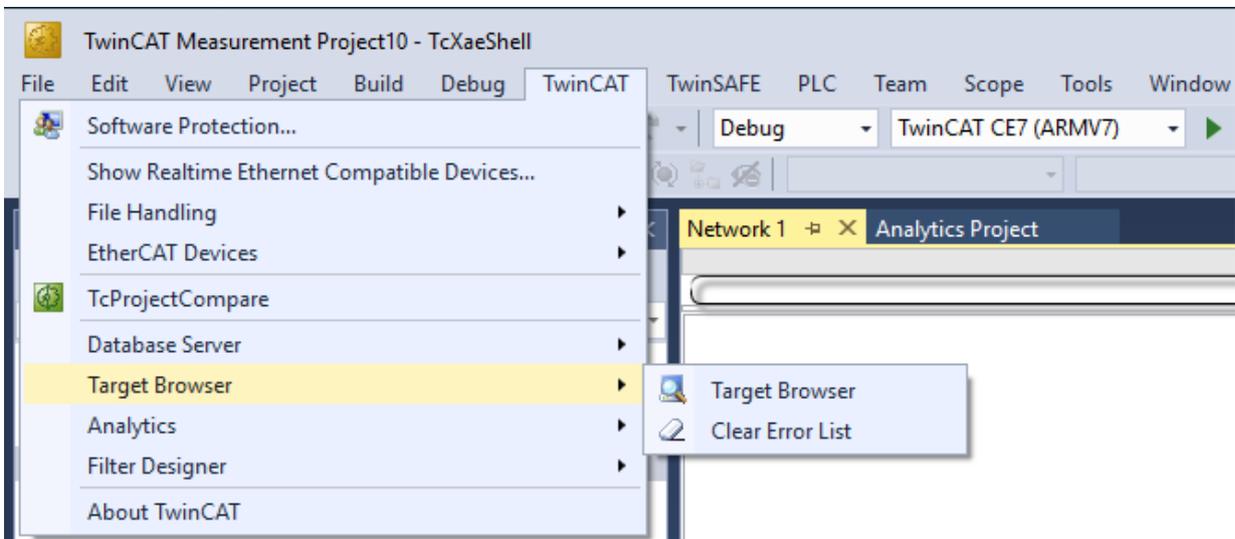
⇒ Das neue Projekt wird im Projektmappen-Explorer angezeigt. Nach einem Klick auf das Baumknotenelement **Analytics Project** öffnet sich ein Startfenster, in dem Sie Ihre erste Aktion auswählen können. Von hier aus können Sie ein Netzwerk hinzufügen, die **Toolbox** öffnen, den **Target Browser** öffnen 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.

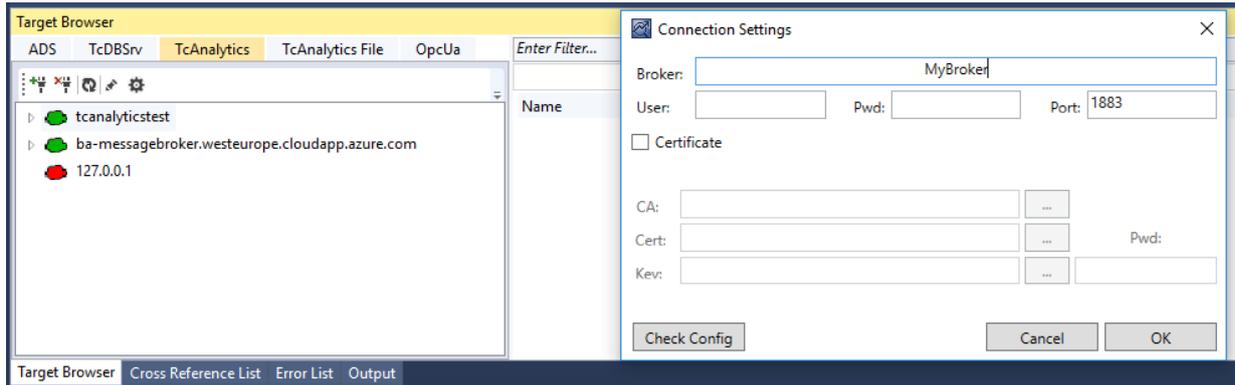


- Wenn Sie auf das Netzwerk klicken, wird ein Editor geöffnet. Nun können Sie den gewünschten Algorithmus per Drag-and-Drop in die Editor-Oberfläche ziehen.
- Nach der Auswahl des Algorithmus müssen Sie Eingangsvariablen mit den Modulen (Algorithmus) verbinden. Öffnen Sie dazu den **Target Browser**.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

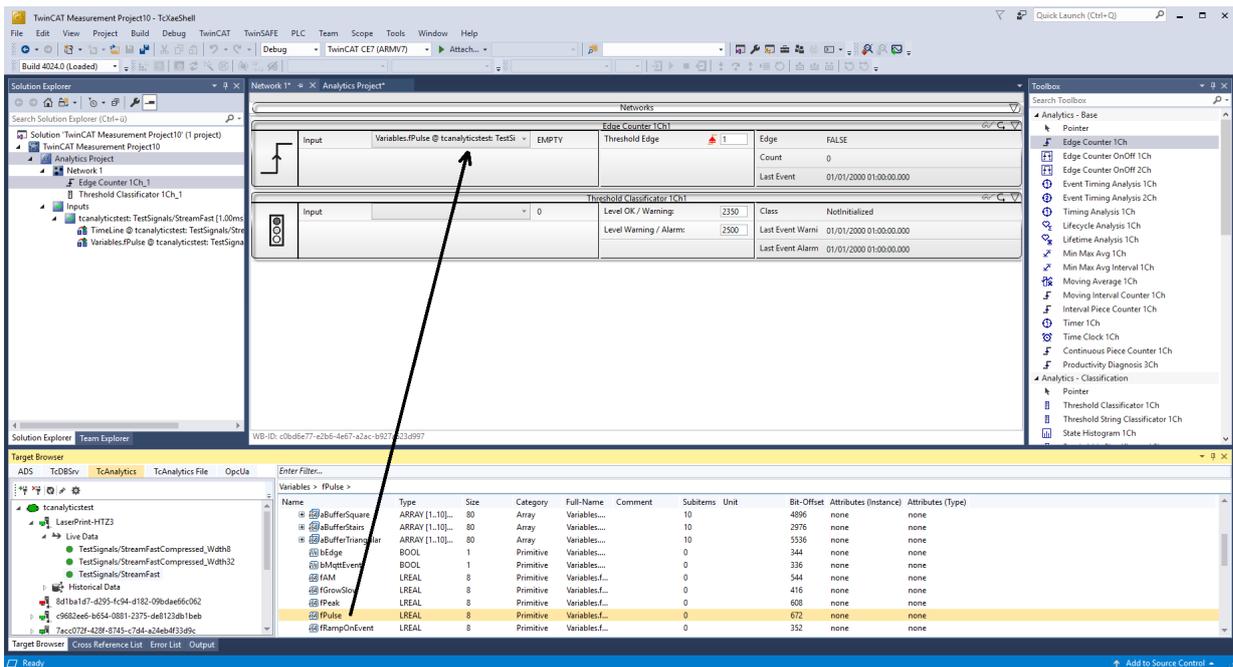
7. Klicken Sie auf das grün markierte Icon in der Symbolleiste dieser Analytics-Erweiterung. Es öffnet sich ein Fenster, in dem Sie die Konnektivitätsdaten Ihres Message-Brokers angeben können.



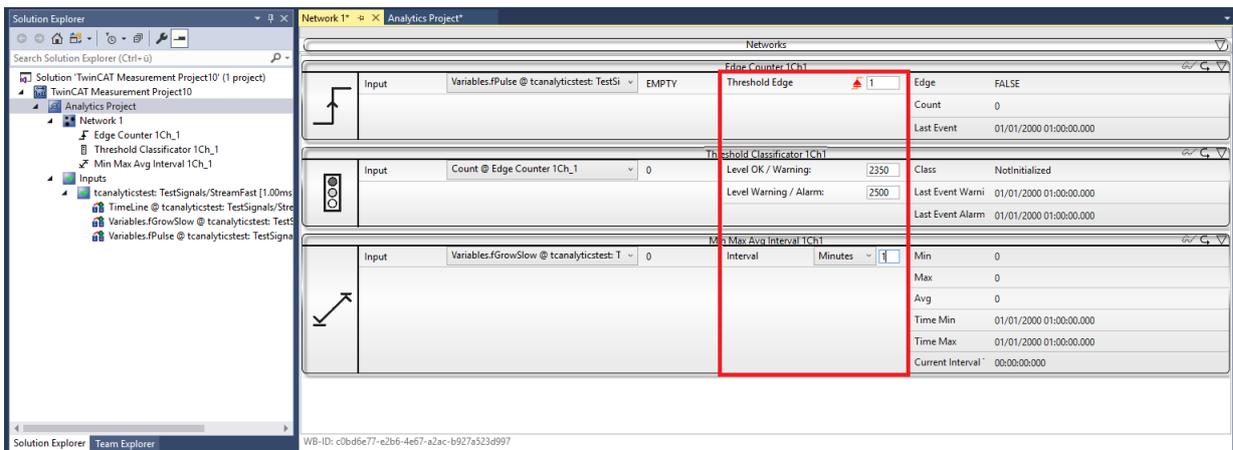
8. Wählen Sie Ihren MQTT-Analytics-Client aus (TwinCAT Analytics Logger, TwinCAT IoT Data Agent oder Beckhoff EK9160). Für jede Steuerung gibt es eine eindeutige ID. Diese ID wird im Target Browser angezeigt.
9. Über einen Klick auf das **Zahnrad-Icon**, gelangen Sie zur Seite Machine Administration.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

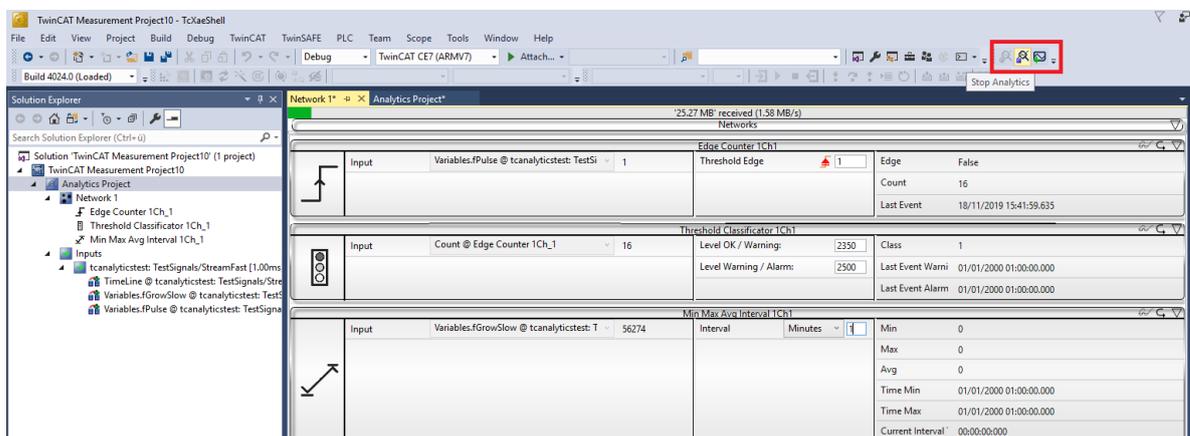
10. Im nächsten Schritt können Sie zwischen Live-Daten und historischen Daten für jeden MQTT Analytics-Client wählen. Die historischen Daten werden in diesem Fall von dem TwinCAT Analytics Storage Provider bereitgestellt.



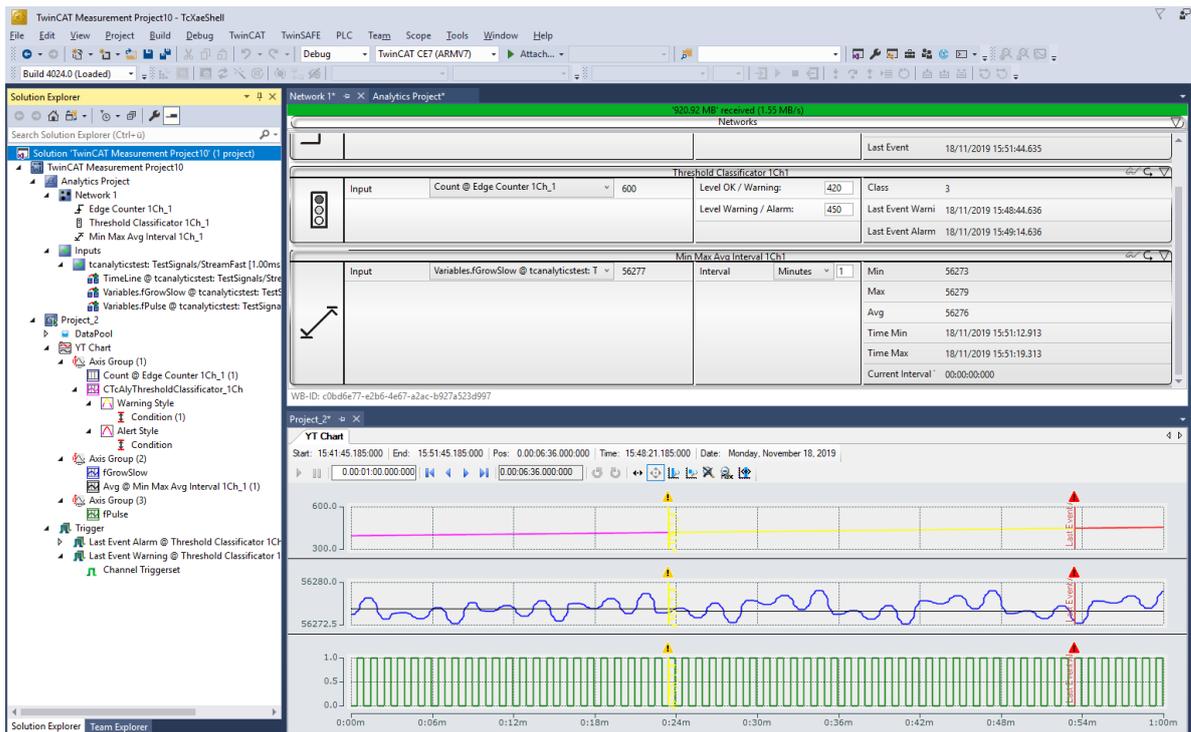
11. Sie können die Variablen per Drag-and-Drop in die Eingänge des spezifischen Algorithmus ziehen. Bei den meisten Algorithmen können Bedingungen wie Schwellen, Zeitintervalle, logische Operatoren usw. festgelegt werden. Diese Einstellungen werden in der Mitte jedes Moduls vorgenommen.



⇒ 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.

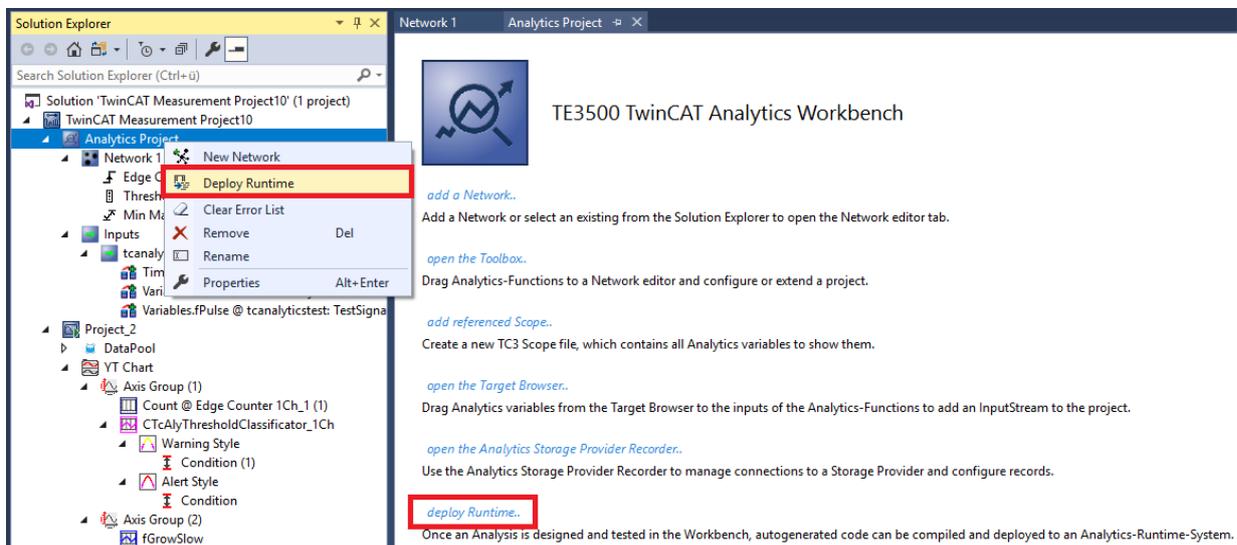


- ⇒ Die Ergebnisse der Analyse können in den Scope View-Grafiken per Drag-and-Drop angezeigt werden. Beispielsweise kann ein Mittelwert als neuer Kanal in der Ansicht angezeigt werden. Zeitstempel als Marker auf den X-Achsen zeigen signifikante Werte.

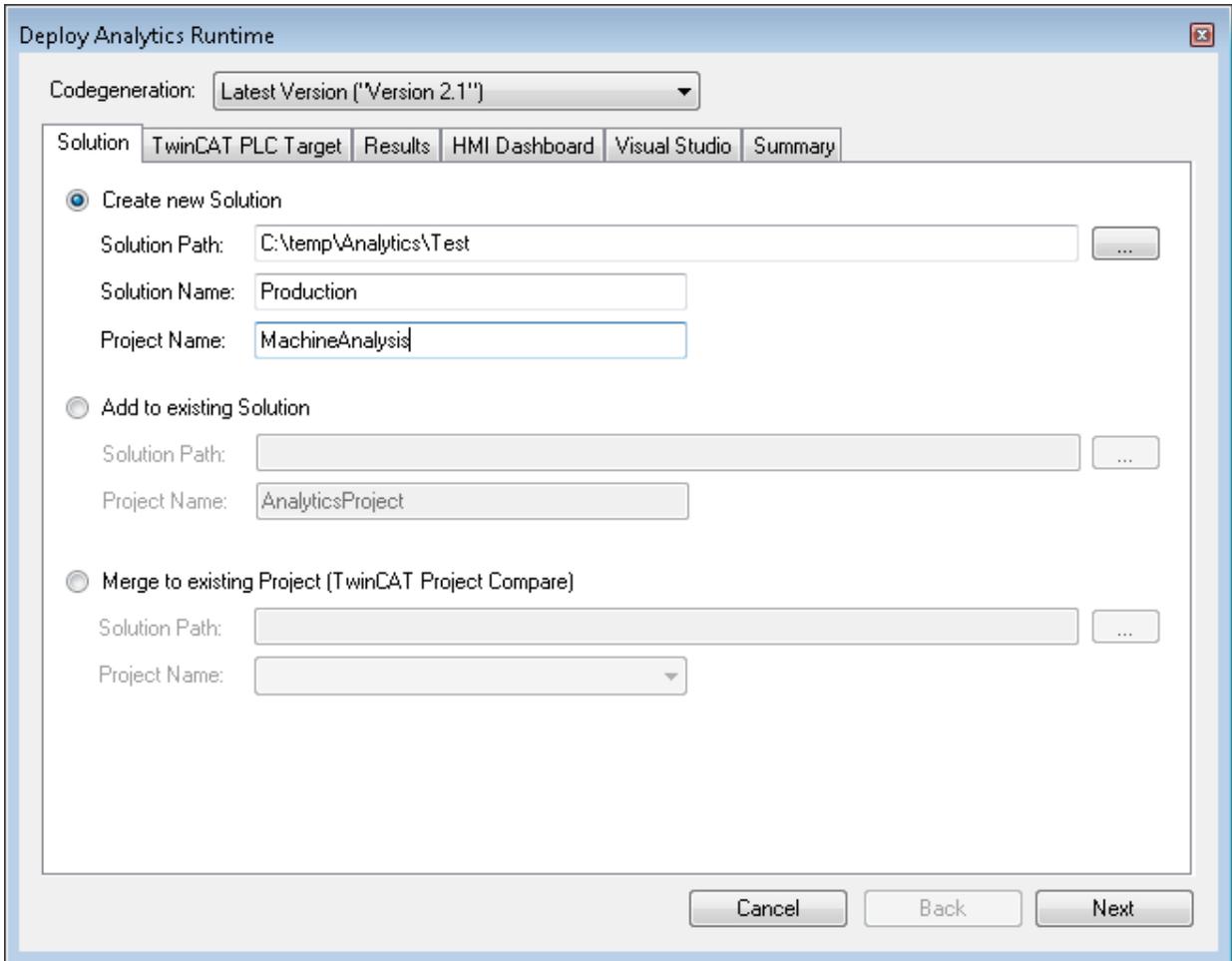
4.6 24-stündige Anwendung von Analytics

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.

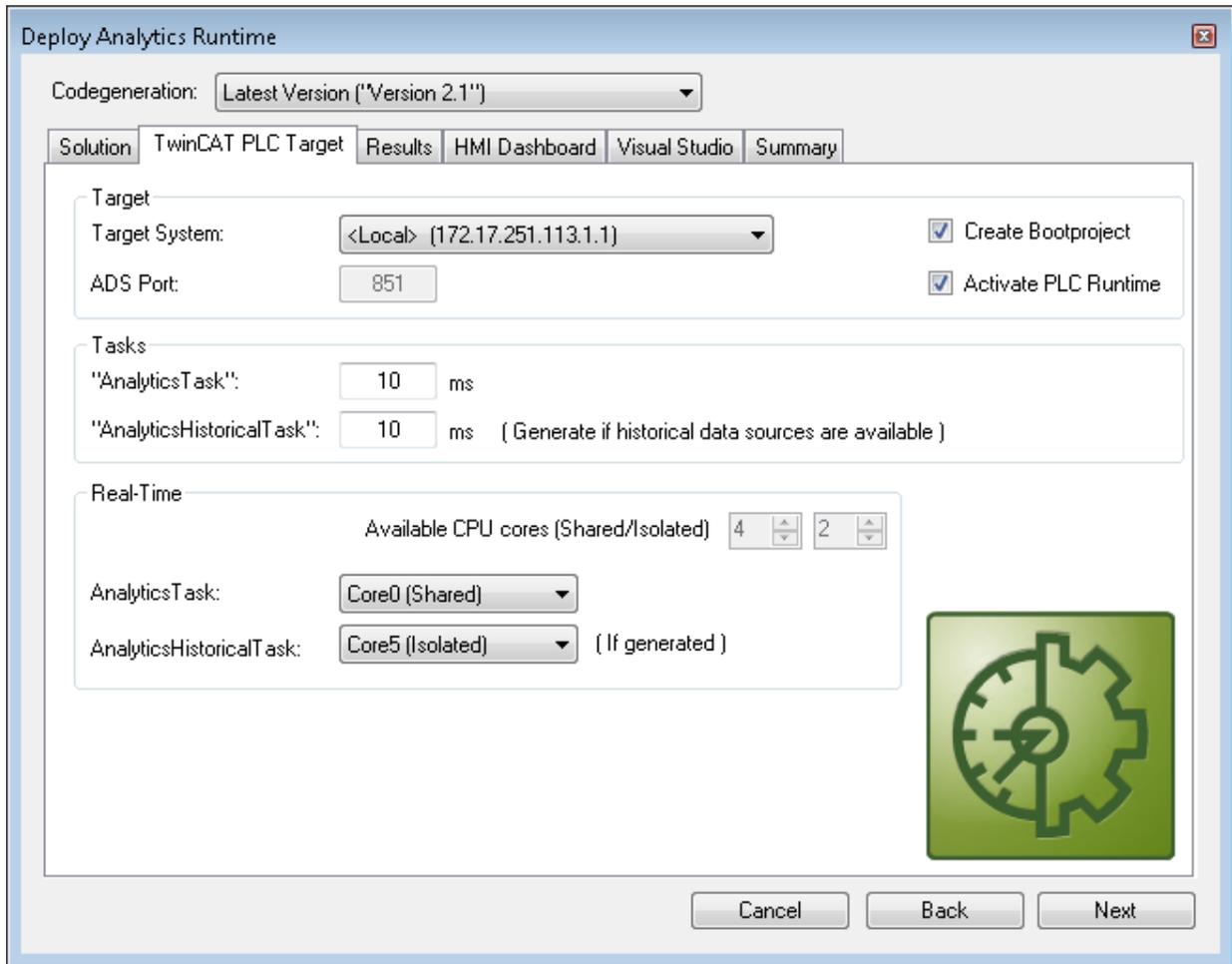
- ✓ Speichern Sie zunächst Ihre Konfiguration und öffnen Sie den Analytics Deploy Runtime-Assistenten. Dies kann über das Kontextmenü im Analytics Project-Tree Item oder über die Startseite erfolgen.



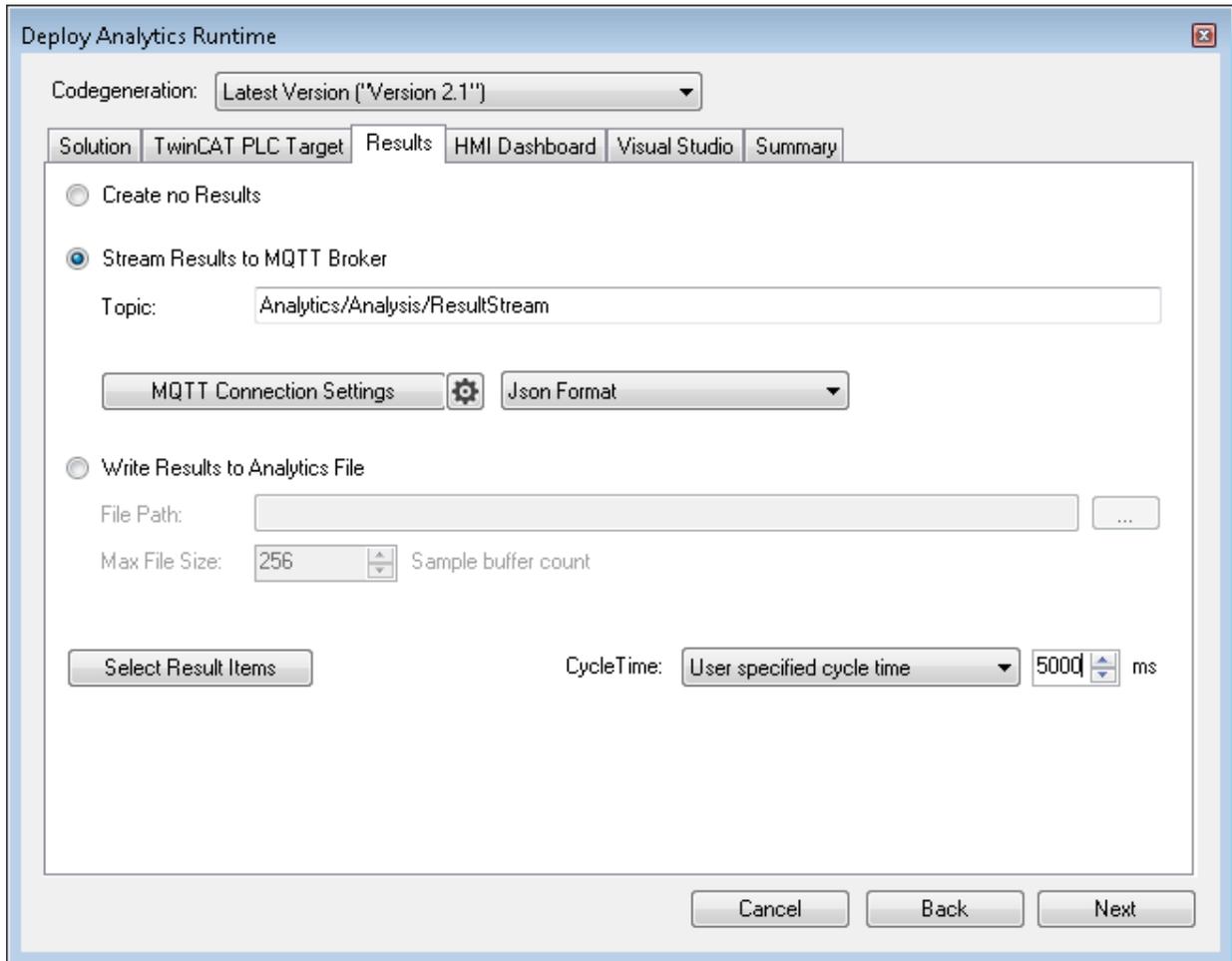
1. Wenn der Assistent geöffnet ist, können Sie sich durch einige Karteireiter klicken. Der erste heißt Solution. Hier können Sie entscheiden, wie Ihr Analytics-Projekt im SPS-Code verwendet werden soll:
 Als...
 vollständig neue Lösung.
 Teil einer vorhandenen Lösung.
 Aktualisierung einer vorhandenen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



- 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.

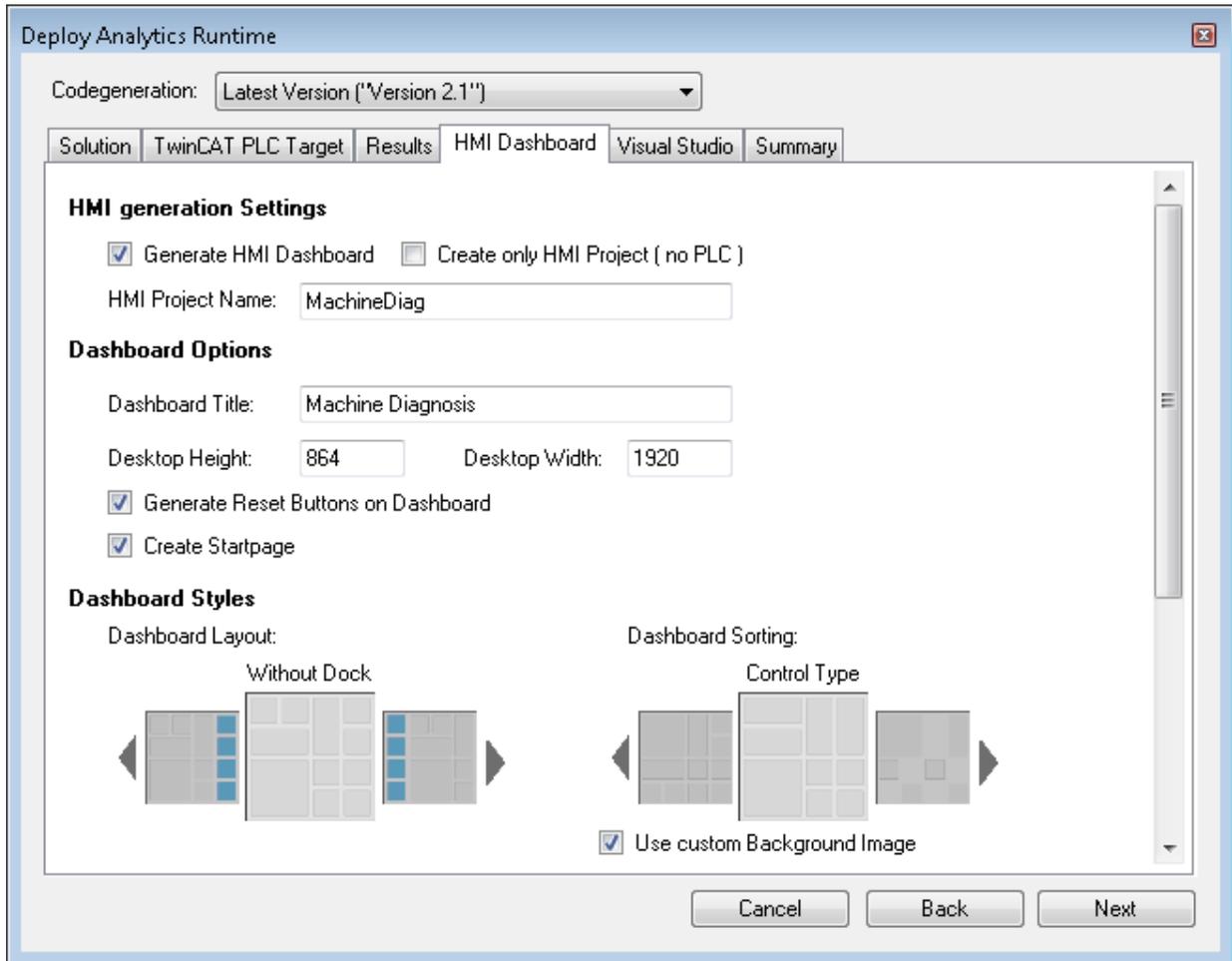


- Insbesondere bei virtuellen Maschinen ist es wichtig, das Projekt auf isolierten Kernen auszuführen, was ebenfalls eine Option in diesem Karteireiter ist. Der nächste Karteireiter **Results** wird nur benötigt, wenn Sie in den Algorithmeigenschaften die Option **Stream Results** ausgewählt haben. Wenn 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.

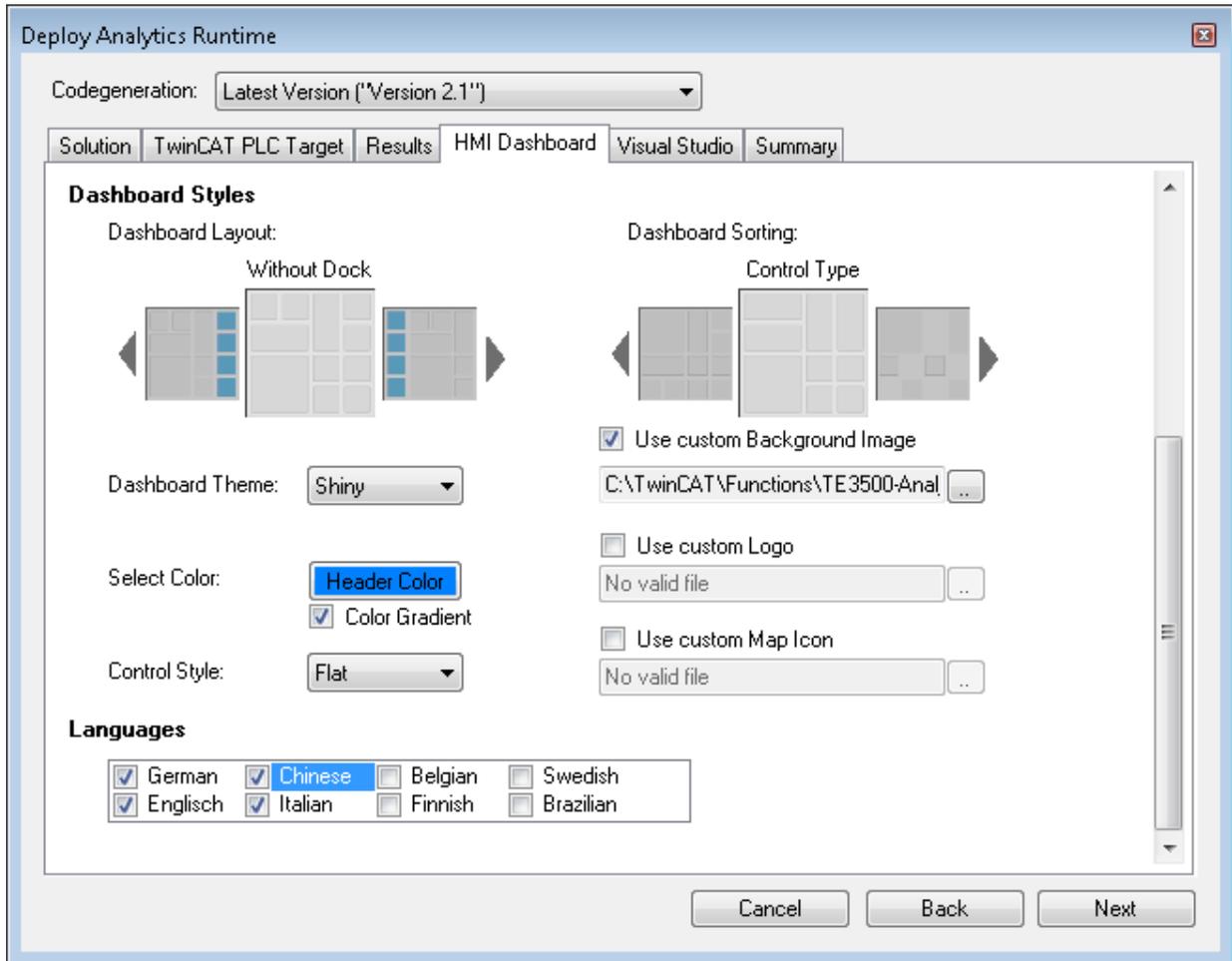


Ein Downsampling der Ergebnisse ist durch die Festlegung einer Zykluszeit möglich. Der nächste Karteneiter ist für das **HMI-Dashboard** vorgesehen. Voraussetzung für die automatische Generierung

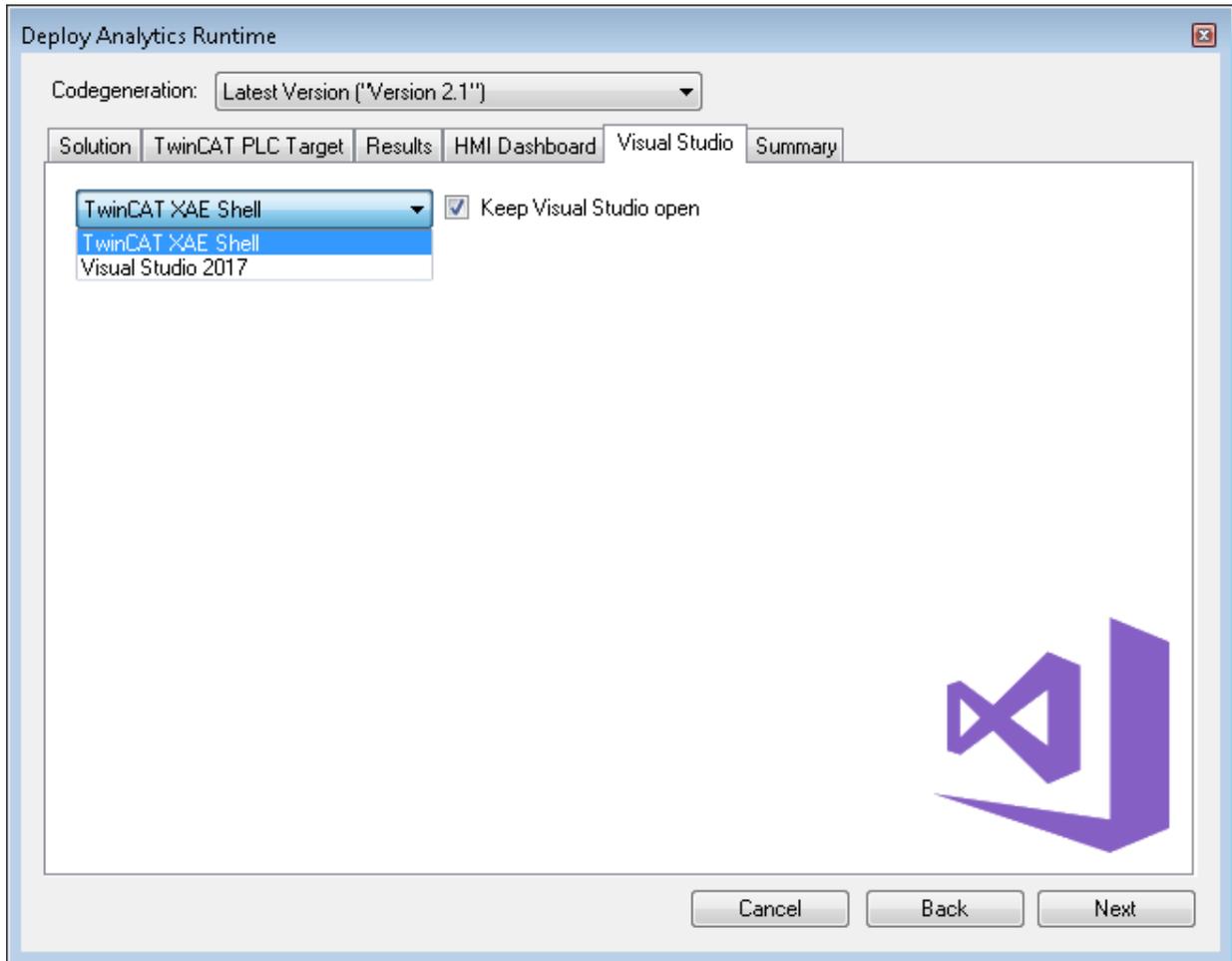
des Dashboards ist die Auswahl von HMI Controls für die entsprechenden Algorithmen, deren Ergebnisse im Dashboard angezeigt werden sollen.



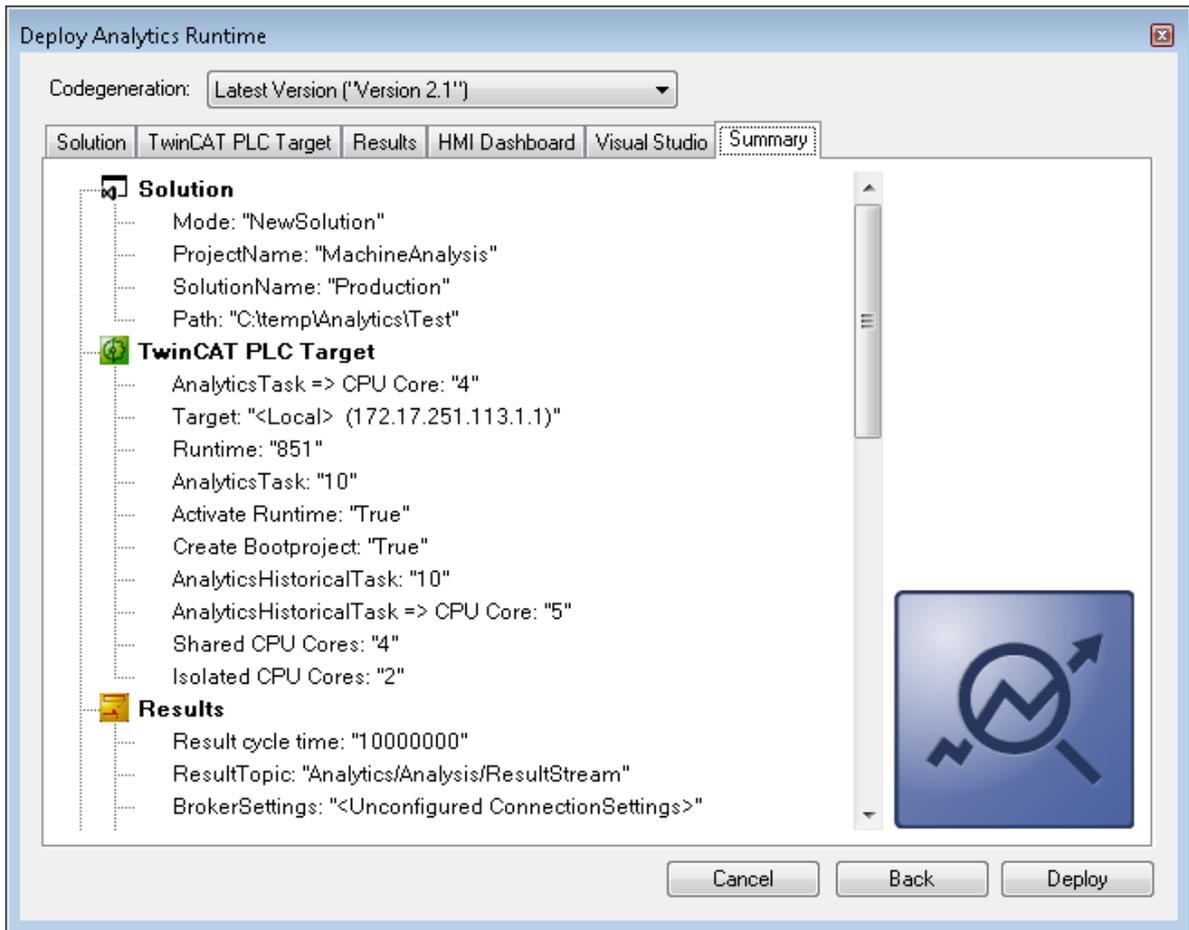
4. Sie können verschiedene Optionen für Ihr Analytics Dashboard auswählen, wie eine Startseite mit einer Karte, Layouts, Sortieralgorithmen, eigene Farben und Logos. Wenn Sie mehrere Sprachen für die Analytics Controls auswählen, wird auch ein Menü zur Sprachumschaltung generiert.



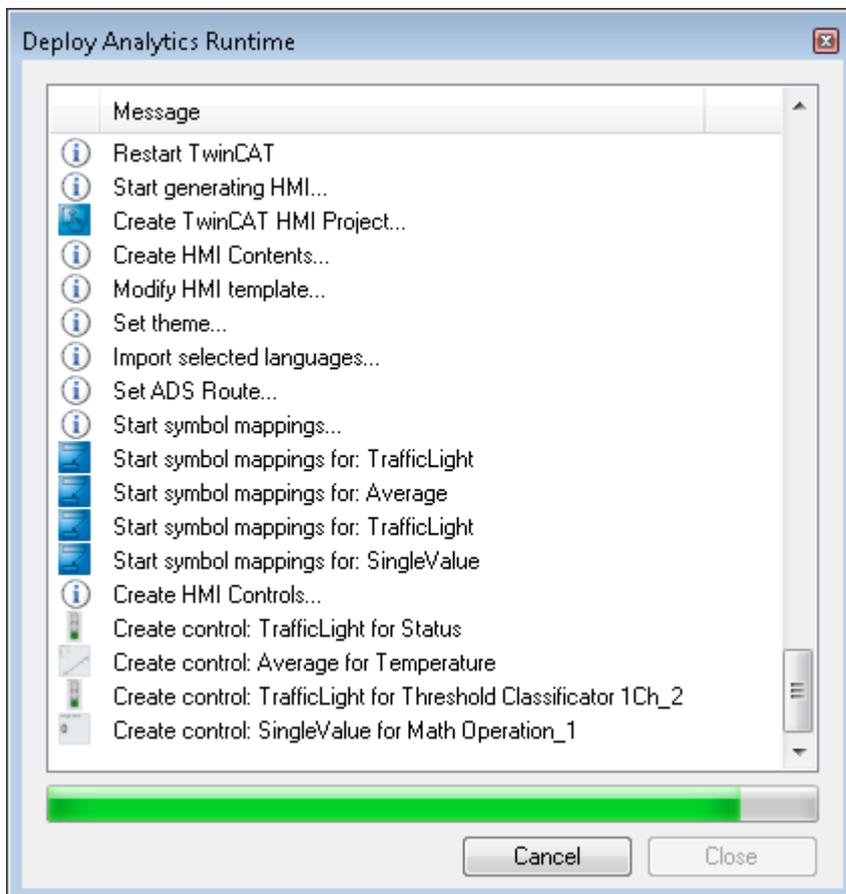
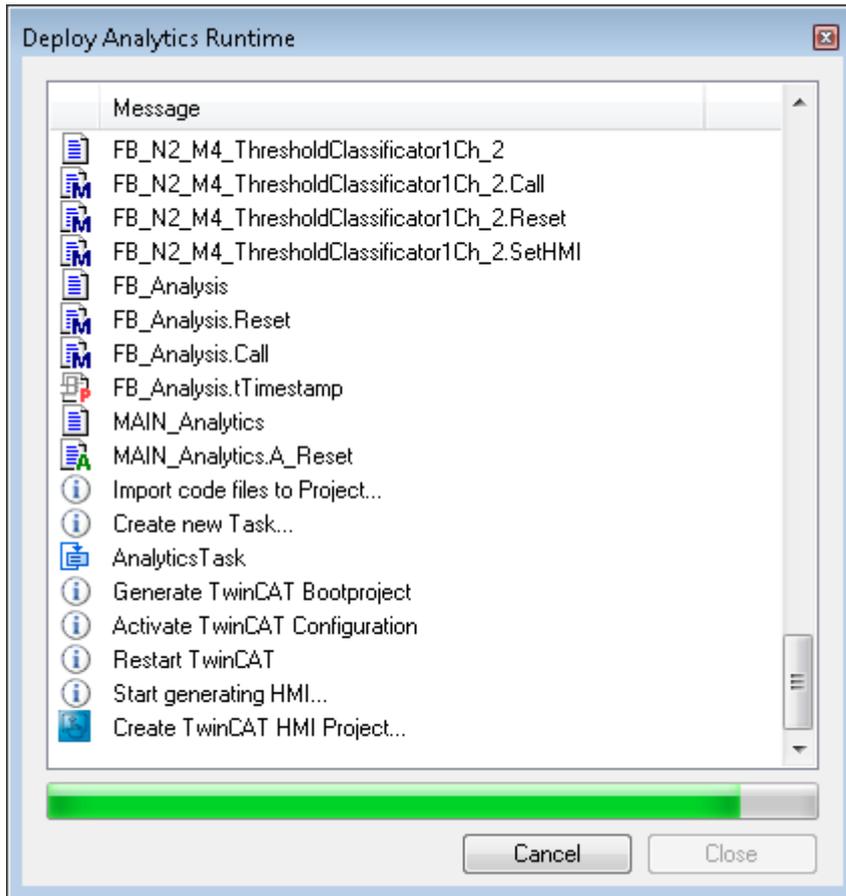
5. Wählen Sie eine der installierten Versionen von Visual Studio® aus, und, ob die Instanz sichtbar starten soll oder ob sie nur eingerichtet und im Hintergrund aktiviert werden soll.



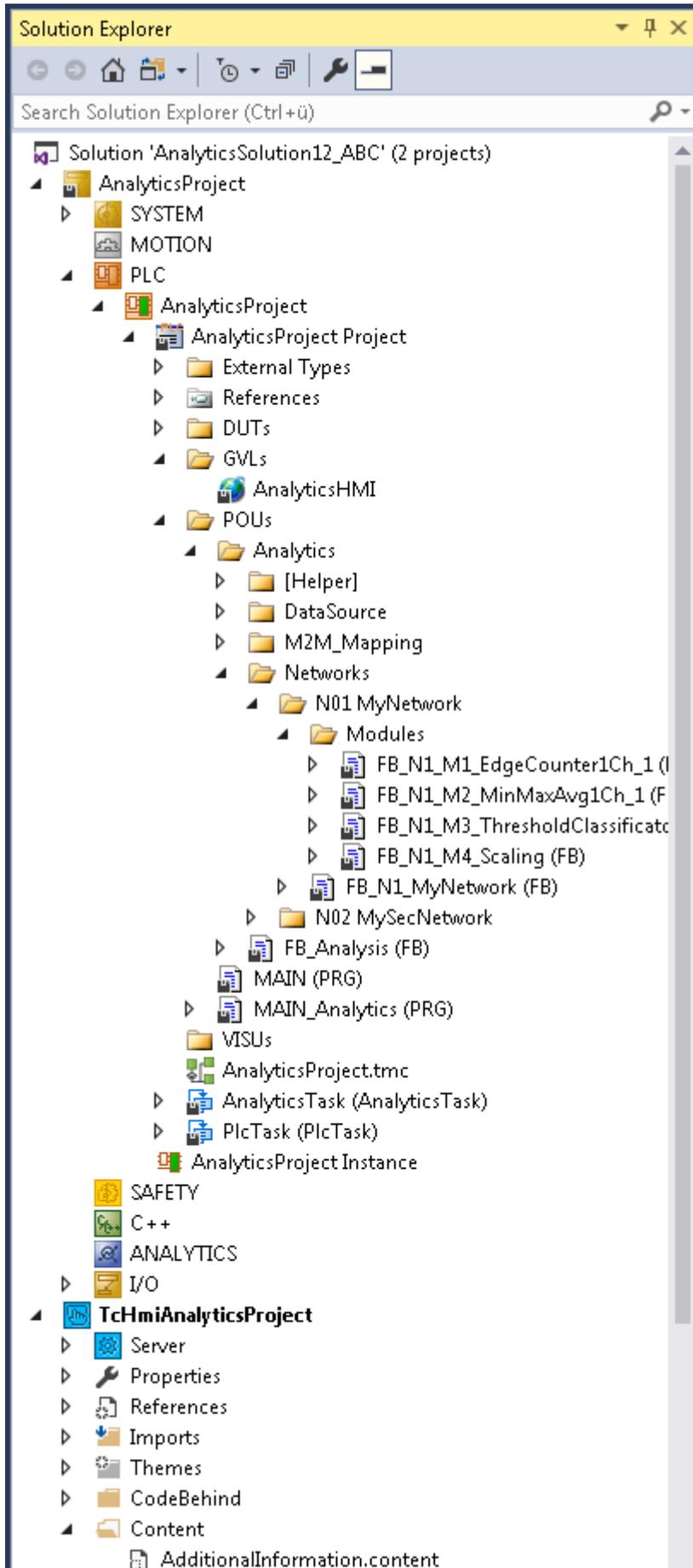
⇒ Zuletzt finden Sie eine Übersicht.



6. 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

Die Grundidee des TwinCAT Analytics Storage Providers (ASP) besteht darin, ein Gateway zu haben, das den Benutzer von der Konfiguration einer Datensinke, also einem Storage bzw einer Datenbank, weitgehend befreit. Der Benutzer muss keine eigene Tabellenstruktur in einer Datenbank einrichten. Er muss lediglich konfigurieren, welche der unterstützten Datensinken er für die Ablage seiner Daten nutzen möchte.

Service-Management

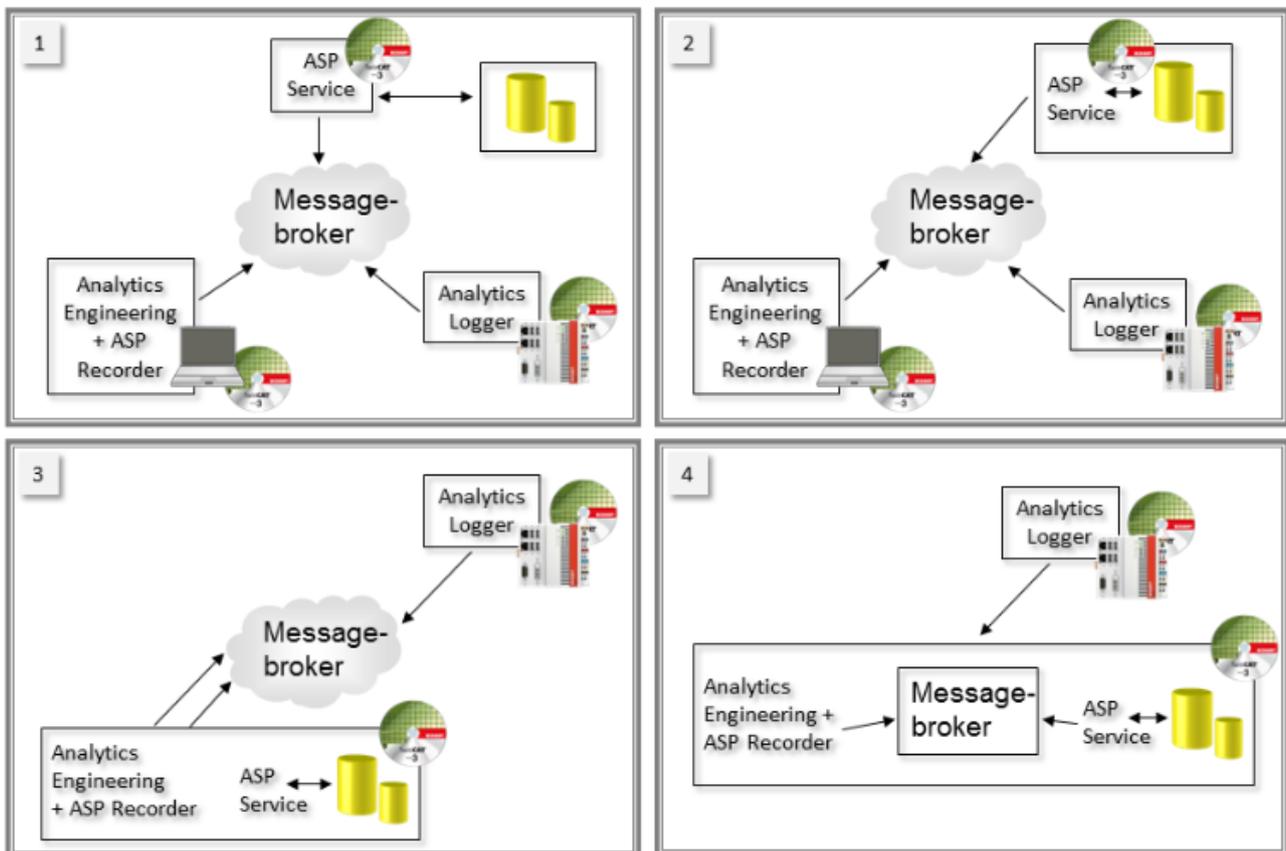
Der Analytics Storage Provider Service kann irgendwo im Netzwerk ausgeführt werden. Er ist als Windows Service implementiert. Der Dienst kann auf Hardware-Geräten, wie Industrie-PCs oder Embedded-PCs im lokalen Netzwerk, und auch auf virtuellen Maschinen im selben Netzwerk, oder z. B. in einem Cloud-System ausgeführt werden.

Daten-Management

Der Storage Provider arbeitet mit dem Binärformat von TwinCAT Analytics. Dadurch kann er Streams von einem MQTT-Message-Broker empfangen und speichern sowie selbst neue Streams erstellen und senden. Der Benutzer benötigt dazu lediglich den Recorder, der mit der TwinCAT Analytics Workbench oder dem Service Tool in seinem eigenen Engineering-System integriert wird. Die Variablen selbst werden im TwinCAT Target Browser angezeigt. Für das Analytics-Binärformat werden sie in Live- und historische Daten unterteilt. Live-Daten können als Eingang für den Analytics Storage Provider verwendet werden. Historische Daten sind die Werte aus der Datenbank/dem Storage, die bzw. der vom Storage Provider bereitgestellt wird.

Topologien

Die vielen Freiheitsgrade der IoT-Technologien ermöglichen verschiedene Topologien. Das folgende Bild zeigt die wichtigsten Konstellationen.



1. Jedes SW-Paket läuft auf einem eigenen HW-Gerät oder einer virtuellen Maschine.
2. Der Windows Service des Analytics Storage Providers läuft auf demselben Gerät wie die Datenbank/ der Storage.

3. Analytics Engineering, Analytics Storage Provider und Datenbank oder Storage befinden sich auf demselben Gerät. Nur der Message-Broker und der Analytics Logger (Datenquelle) laufen auf anderen Geräten.
4. In dieser Topologieansicht läuft nur der Analytics Logger auf einem eigenen PC. Dies kann in einer Maschinenanwendung der Fall sein. Alle anderen Tools der Analytics-Toolkette befinden sich auf einem Gerät, einschließlich des MQTT-Message-Brokers.

Topologien mit zusätzlichen ASP Clients

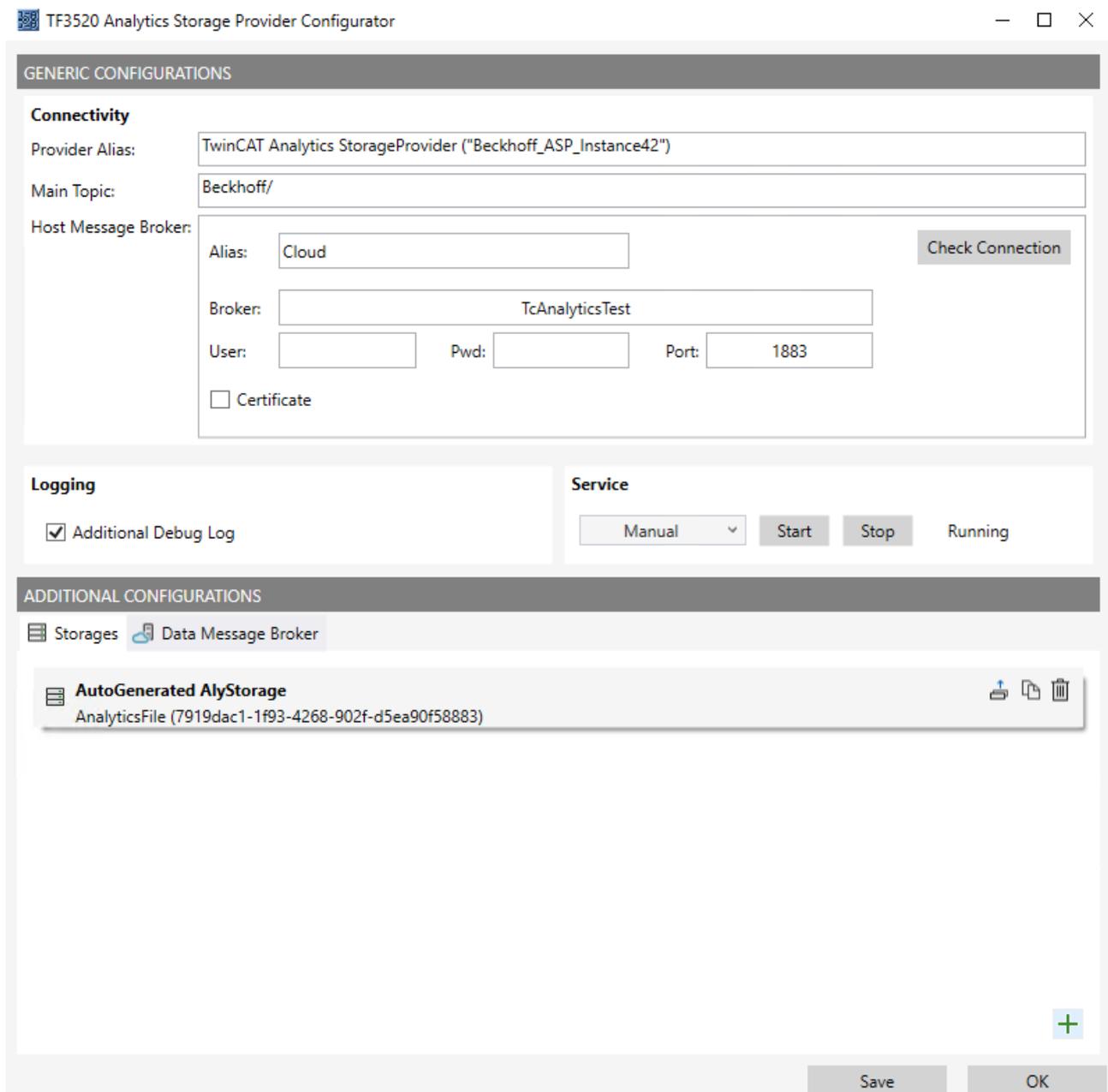
Aktuell sind zwei weitere Clients aus Sicht des Analytics Storage Providers verfügbar. Ein Kommandozeilenbasierter Client, der eine Ausführung aus nahezu jeder Applikation zulässt. Und eine SPS Bibliothek mit der ebenfalls Einfluss auf die Aktionen des Storage Providers genommen werden kann.

6 Konfiguration

Die Konfiguration des Analytics Storage Providers ist in zwei Hauptteile unterteilt. Zum einen müssen Sie den Service mit seinen Stores konfigurieren. Das passiert im sogenannten TwinCAT Analytics Storage Provider Configurator. Des Weiteren müssen Sie die Aufnahmen und Pipelines selbst konfigurieren. Also welche Variablen sollen unter welchen Bedingungen in welchem Stores abgelegt werden. Dafür gehen Sie in den TwinCAT Analytics Storage Provider Manager. In diesem Kapitel finden Sie auch die unterstützten Datenbanken und Storages.

6.1 Configurator

Im TwinCAT Analytics Storage Provider Configurator können Sie den Service mit seinen Message Brokern und Stores konfigurieren. Dabei wird zwischen einem Host Message Broker und verschiedenen Data Message Brokern unterschieden. Ein Host Message Broker kann gleichzeitig auch ein Data Message Broker sein. Der Host Message Broker ist insofern besonders, da sich hier der Service des Storage Providers anmeldet.



6.1.1 Generic Configurations

In den Generic Configurations können allgemeine Einstellungen zum Service des Storage Providers getätigt werden.

GENERIC CONFIGURATIONS

Provider Alias:

Jeder Analytics Storage Provider Service hat eine eigene GUID zur Identifikation. Damit diese durch einen sprechenden Namen ersetzt werden kann, können Sie in dem allgemeinen Konfigurationsbereich einen Provider Alias angeben.

Main Topic:

Grundsätzlich müssen Sie sich nicht um das Topic kümmern. Sie können jedoch hier das sogenannte Main Topic eingeben. Es beschreibt den ersten Teil des Gesamt-Topics. Beckhoff spezifische Ergänzungen werden danach angehängt. So ergibt sich für Sie ein sehr einfaches Plug-and-play System mit TwinCAT Analytics.

Host Message Broker:

Hier kann der sogenannte Host Message Broker konfiguriert werden. An diesem Message Broker meldet sich der Storage Provider Service selbst an und stellt seine Daten zur Verfügung. Dieser Broker kann gleichzeitig ein sogenannter Data Message Broker sein, der zum Empfang von Daten an den Storage Provider Service dient. Es kann im System nur einen Host Broker, dafür aber mehrere Data Broker geben.

Logging:

Haken Sie verschiedene Logging-Optionen an.

Service:

Anzeige des Status und der Standardeinstellungen zum Starten des Windows Service.

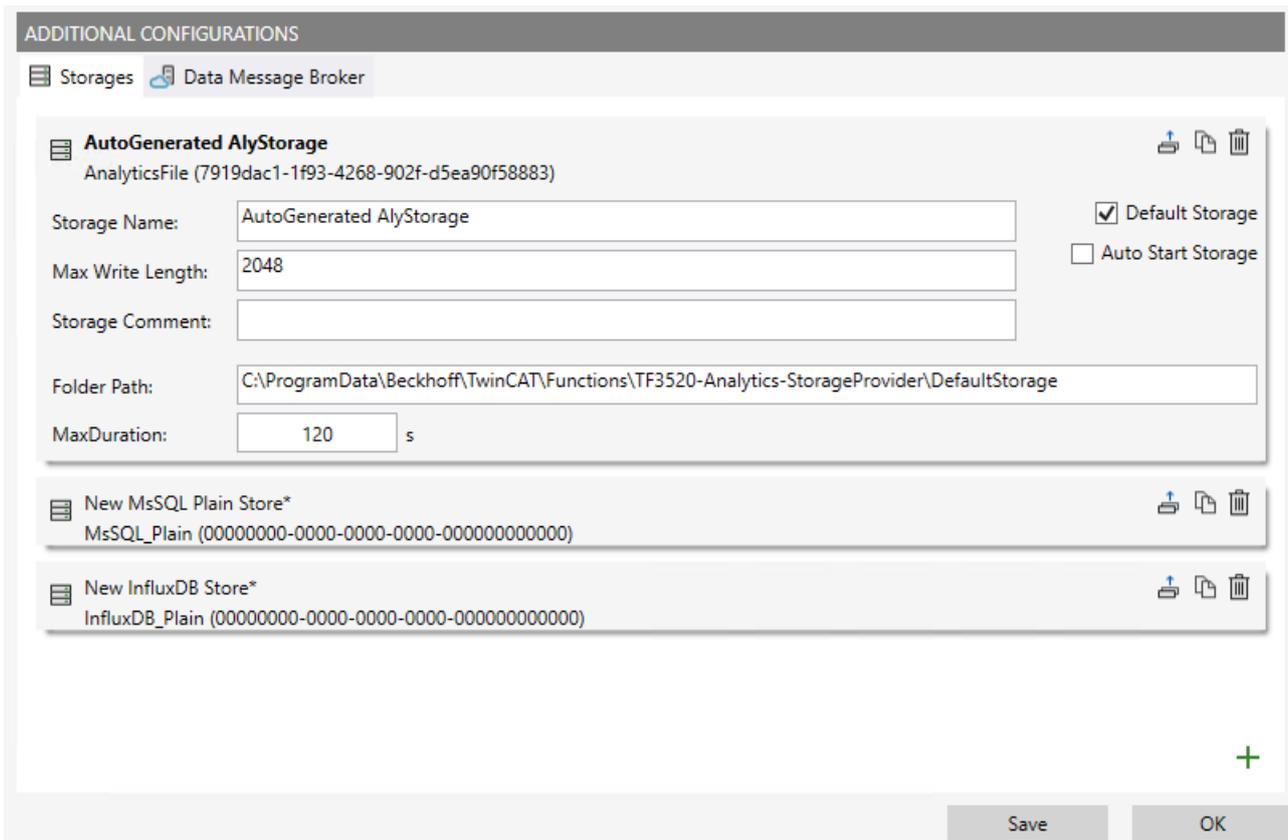
6.1.2 Additional Configurations

Die Stores und Data Message Broker des Storage Provider Services werden unter Additional Configurations eingestellt.

ADDITIONAL CONFIGURATIONS

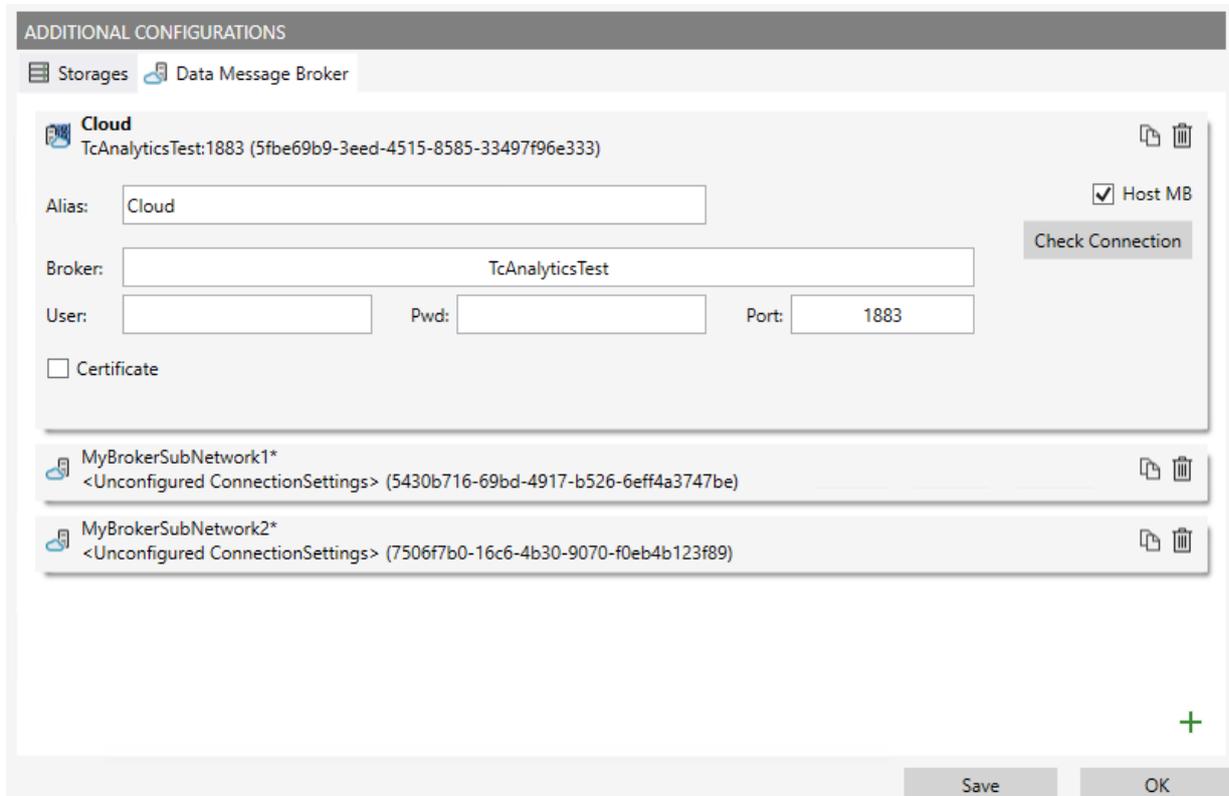
Storages

Im Tab Storages können die Stores angelegt werden, die dem Analytics Storage Provider Service zur Verfügung stehen. Das können unterschiedliche sein wie Microsoft SQL® oder csv Dateien. Es können über das Plus-Zeichen in der unteren rechten Ecke neue Stores im Binär- oder Klartext-Format hinzugefügt werden. Die Konfiguration der einzelnen Stores wird in diesem [Unterkapitel \[▶ 50\]](#) beschrieben.



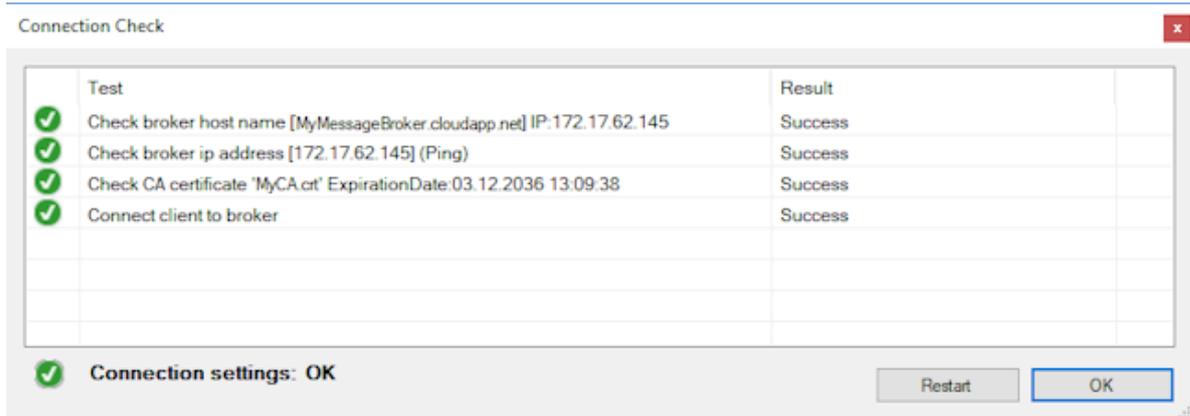
Data Message Broker

1. Im Tab Data Message Broker können die Message Broker angelegt werden, die zusätzlich zum Host Message Broker Input-Daten für den Storage Provider liefern können.



2. Die Einstellungen können mit dem Check Connection Button überprüft werden.

⇒ Das Ergebnis wird im folgenden Fenster angezeigt:



Wenn Sie auf **Save** klicken, werden die Einstellungen im Verzeichnis *C:\TwinCAT\Functions\TF3520-Analytics-StorageProvider\Configurations* gespeichert.

6.1.2.1 Datenbanken/Storage

Die nachfolgende Übersicht zeigt, welche Datenbankverbindungen von welcher Plattform unterstützt werden.

Database	Windows		TwinCAT/BSD	
	Local	Remote	Local	Remote
Analytics File	X	X	X	X
CSV	X	X	X	X
MS SQL	X	X	-	X
PostgreSQL	X	X	-	X
InfluxDB 2	X	X	-	X
AzureBlob	-	X	-	X

6.1.2.1.1 TwinCAT Analytics Binary File

TwinCAT Analytics Binary File ist ein TwinCAT-spezifischer Storage. Daher ist keine externe Software erforderlich. Sie können diese Speicherart direkt nach der Installation des Analytics Storage Providers nutzen. Es handelt sich um dieselbe Datei, die der TwinCAT Analytics Logger in seiner „Offline“-Konfiguration ohne MQTT-Message-Broker bereitstellt.



Abb. 1:

Storage Name:

Vergeben Sie einen sprechenden Namen, der den Zweck des Storage gut beschreibt. Er wird bei der Konfiguration im Manager an verschiedenen Stellen wieder erscheinen.

Max Write Length:

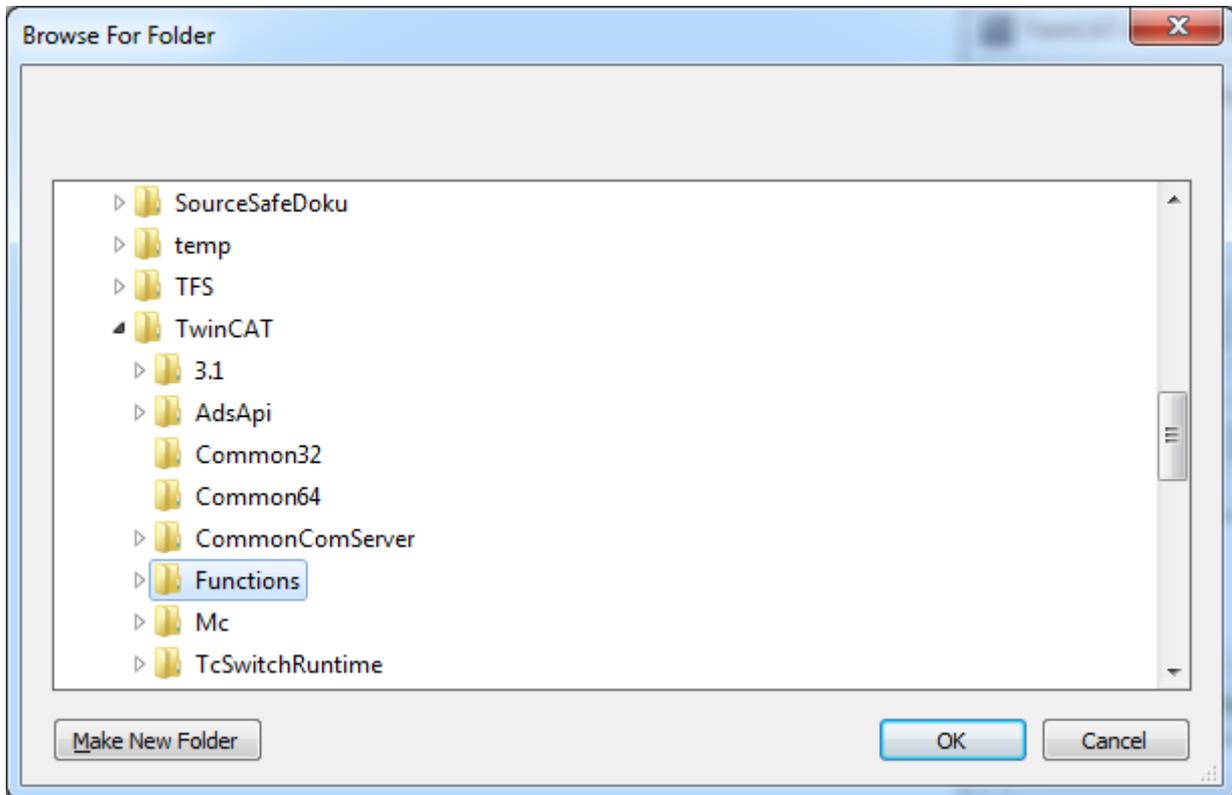
Hier wird die Datenmenge angegeben, welche in einem Aufruf in die .tay-Datei geschrieben wird.

Storage Comment:

Hier können Sie zusätzliche Informationen zu dem Storage hinterlegen.

File Path:

Für die Konfiguration müssen Sie Ihren bevorzugten Ordner auf dem lokalen Gerät auswählen, auf dem der Storage Provider läuft.



Zur Bestätigung wird der verwendete Ordner im Fenster Connection String angezeigt.

Die später nach dem Start der Pipeline erstellte Ordner-Architektur gliedert sich aktuell in fünf Hierarchieebenen:

1. System-ID (das ist eine GUID des Systems, welches die Daten sendet) - kann durch den Alias Namen in den Analytics Logger Einstellungen oder im lokalen Engineering durch die Machine Administration Page ersetzt werden.
2. Recording-Name – kann durch den Konfigurator eingestellt werden.
3. Record-Name – kann durch den Konfigurator eingestellt und mit Auto-IDs, Datum, Zykluszeit und anderen Platzhaltern zur Laufzeit automatisch beeinflusst werden.
4. Interne ID – kann nicht verändert werden.
5. Layout GUID – kann nicht verändert werden und entspricht der Datenablage des Analytics Loggers.

Max Duration:

Dieser Wert gibt in Sekunden an wie viele Daten in ein .tay-File geschrieben werden. Nach der Zeit X werden die Daten in eine neue .tay-Datei geschrieben. Ringpuffer die im Manager konfiguriert werden, wirken sich ausschließlich auf vollständige .tay-Files eines aktuellen Recordings aus.

6.1.2.1.2 Microsoft SQL (Binär / Klartext)

Mit dem Microsoft SQL Server haben Sie eine weitere On-Premises-Lösung für die Speicherung der Analytics-Binärdaten.

Geben Sie hier den Connection String für Ihren MS SQL-Server an.

My MsSQL Plain Store*
MsSQL_Plain (00000000-0000-0000-0000-000000000000)

Storage Name: My MsSQL Plain Store Default Storage

Max Write Length: 1024 Auto Start Storage

Storage Comment: First MsSQL Plain Store (MyServer)

Connectionstring: Data Source=MyServer\SQLEXPRESS; Initial Catalog=TcAlyStorageProviderDB; User Id='ASP_User'; Password='*****'

ConnString

Storage Name:

Vergeben Sie einen sprechenden Namen, der den Zweck des Storage gut beschreibt. Er wird bei der Konfiguration im Manager an verschiedenen Stellen wieder erscheinen.

Max Write Length:

Hier wird die Datenmenge angegeben, welche in einem tbl_Data Datensatz gespeichert wird.

Storage Comment:

Hier können Sie zusätzliche Informationen zu dem Storage hinterlegen.

Connection String:

Klicken Sie auf die Schaltfläche **ConnString**, um die Eingabemaske zu öffnen. Nehmen Sie dort die Konfigurationseinstellungen vor, auch von Remote-Datenbanken, die durch Netzwerkverbindungen erreichbar sind.

MsSQL ConnectionString

Server: MyServer\SQLEXPRESS

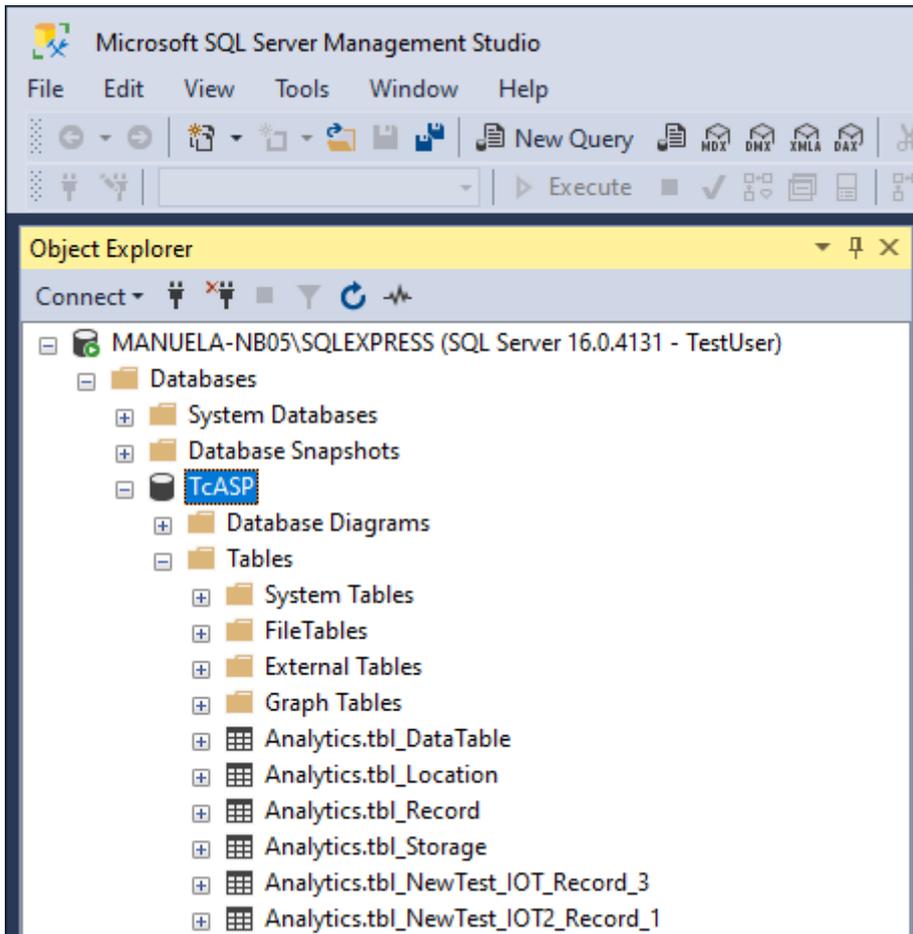
Database: TcAlyStorageProviderDB

Username: ASP_User

Password: *****

Save Close

Nach dem Start des Storage beginnt die Kommunikation mit der Datenbank. In diesem Moment erstellt der Storage Provider die vier notwendigen Tabellen selbst. Jede Aufnahme-Konfiguration wird in einer eigenen Tabelle gespeichert. Als Beispiel sehen Sie den folgenden Screenshot vom Microsoft SQL Server Management Studio.



Die Ringpuffer Funktionalität für die Microsoft SQL® Datenbank kann ebenfalls im Manager eingestellt werden.

6.1.2.1.3 CSV-Datei



Storage Name:

Vergeben Sie einen sprechenden Namen, der den Zweck des Storage gut beschreibt. Er wird bei der Konfiguration im Manager an verschiedenen Stellen wieder erscheinen.

Max Write Length:

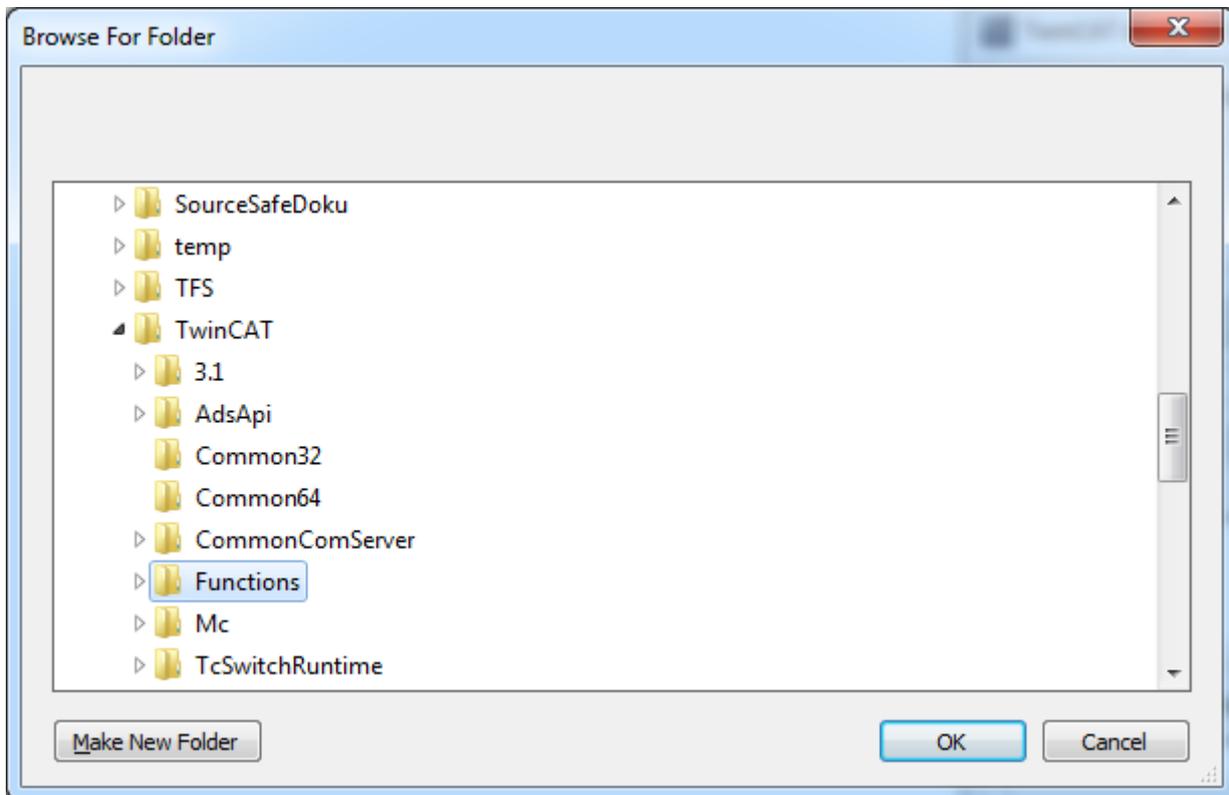
Hier wird die Datenmenge angegeben, welche in einem Aufruf in die .csv-Datei geschrieben wird.

Storage Comment:

Hier können Sie zusätzliche Informationen zu dem Storage hinterlegen.

File Path:

Für die Konfiguration müssen Sie Ihren bevorzugten Ordner auf dem lokalen Gerät auswählen, auf dem der Storage Provider läuft.



Zur Bestätigung wird der verwendete Ordner im Fenster Connection String angezeigt.

Max Duration:

Zusätzlich besteht die Möglichkeit, die Zeitspanne zu definieren, die in einer CSV-Datei gespeichert werden soll. Es können auch die Dezimalstellen von Fließkommazahlen begrenzt werden.

Decimalplaces:

Hier kann die Anzahl der Dezimalstellen eingestellt werden. Der Wert „-1“ steht für nicht spezifiziert, der Wert „2“ steht beispielsweise für zwei Nachkommastellen.

6.1.2.1.4 InfluxDB

Mit der Unterstützung von InfluxDB haben Sie eine weitere On-Premises-Lösung für die Speicherung der Analytics-Binärdaten.

New InfluxDB Store
InfluxDB_Plain (aae02217-13f7-43df-bf52-cbab9e6cf408)

Storage Name: Default Storage

Max Write Length: Auto Start Storage

Storage Comment:

Connectionstring:

Storage Name:

Vergeben Sie einen sprechenden Namen, der den Zweck des Storage gut beschreibt. Er wird bei der Konfiguration im Manager an verschiedenen Stellen wieder erscheinen.

Max Write Length:

Hier wird die Datenmenge angegeben, welche in einem Aufruf in die Influx geschrieben wird.

Storage Comment:

Hier können Sie zusätzliche Informationen zu dem Storage hinterlegen.

Connection String:

Klicken Sie auf die Schaltfläche **ConnString**, um die Eingabemaske zu öffnen. Dort können Sie die Konfigurationseinstellungen vornehmen, auch von Remote-Datenbanken, die durch Netzwerkverbindungen erreichbar sind. Der Server sollte immer mit Port-Nummer angegeben werden.

Nach dem Start des Storage beginnt die Kommunikation mit der Datenbank.

6.1.2.1.5 Microsoft Azure Blob

Um den Storage von Microsoft Azure Blob zu nutzen, benötigen Sie ein Microsoft Azure Cloud-Konto. Dort erhalten Sie auch Ihren individuellen Connection String für die Konfiguration des TwinCAT Analytics Storage Providers.

Storage Name:

Vergeben Sie einen sprechenden Namen, der den Zweck des Storage gut beschreibt. Er wird bei der Konfiguration im Manager an verschiedenen Stellen wieder auftauchen.

Max Write Length:

Hier wird die Datenmenge angegeben, welche in einem Aufruf in den Blob Store geschrieben wird.

Storage Comment:

Hier können Sie zusätzliche Informationen zu dem Storage hinterlegen.

Kopieren Sie den Connection String in das Beschreibungsfeld. Der Speicher muss in Azure selbst erzeugt werden.

Wählen Sie **Storage accounts (classic)** aus.

The screenshot displays the Microsoft Azure portal interface for creating a storage account. The left sidebar shows the navigation menu with 'Storage accounts (classic)' selected. The main content area is titled 'Create storage account' and is divided into two main sections: 'PROJECT DETAILS' and 'INSTANCE DETAILS'.

PROJECT DETAILS:

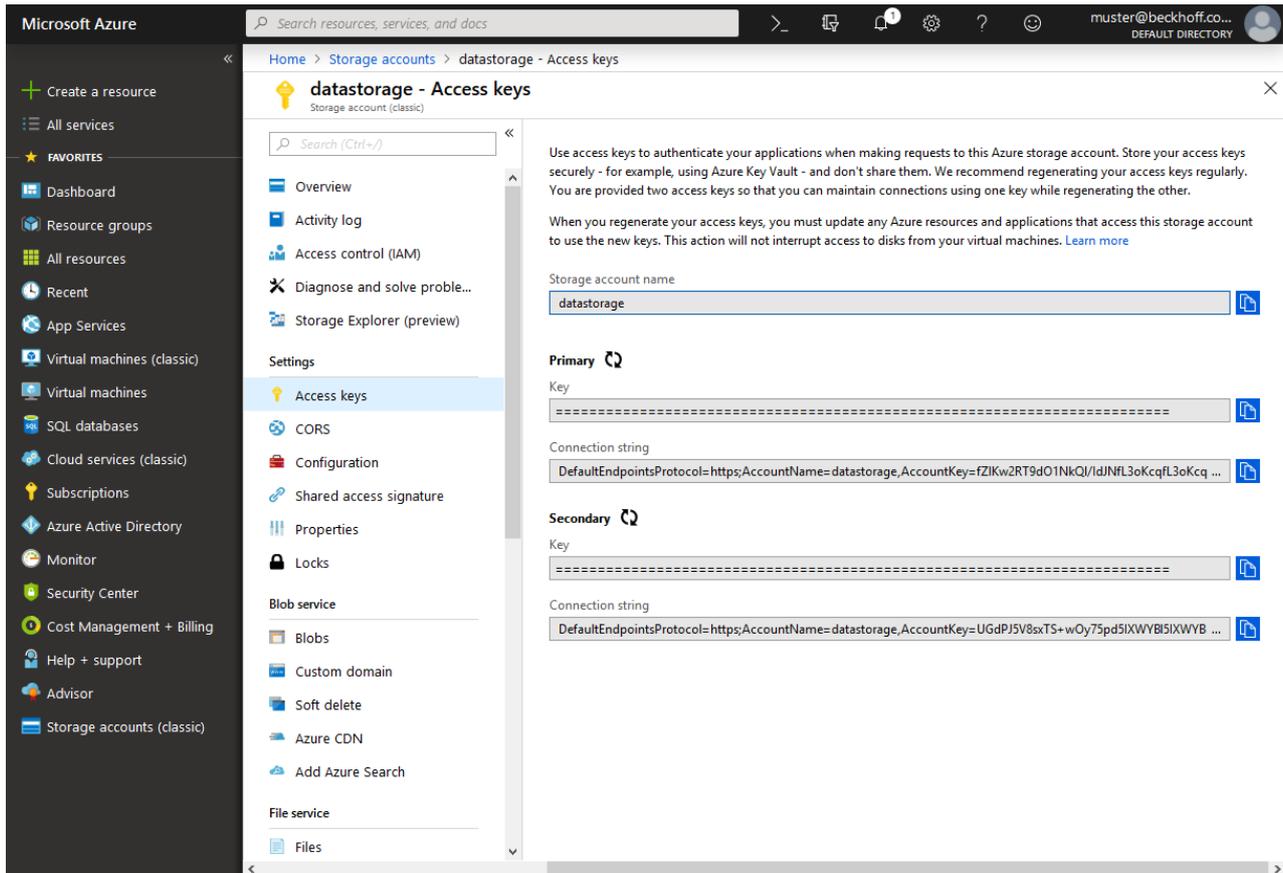
- Subscription:** Visual Studio Professional
- Resource group:** Default-Storage-NorthEurope

INSTANCE DETAILS:

- Storage account name:** (empty field)
- Location:** North Europe
- Performance:** Standard (selected), Premium
- Account kind:** StorageV2 (general purpose v2)
- Replication:** Read-access geo-redundant storage (RA-GRS)
- Access tier (default):** Cool, Hot (selected)

At the bottom of the page, there are three buttons: 'Review + create' (highlighted in blue), 'Previous', and 'Next : Advanced >'.

Nach der Erzeugung des Storage finden Sie unter **Access keys** den sekundären Connection String. Dieser String muss in der Konfiguration des Analytics Storage Providers verwendet werden.



6.1.3 Importierung/Konvertierung von Analytics Files

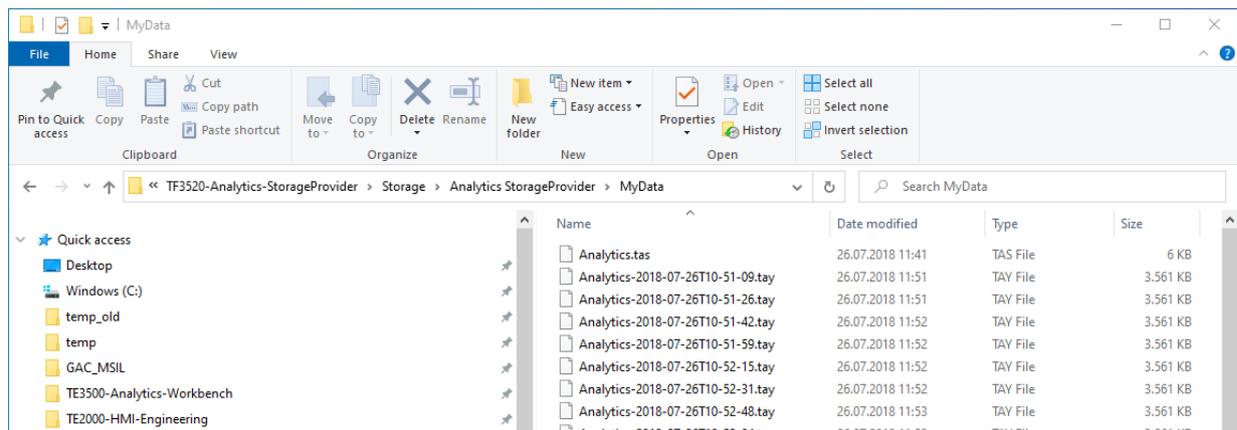


Im Folgenden wird davon ausgegangen, dass Sie TwinCAT unter „C:/TwinCAT“ installiert haben. Andernfalls müssen Sie die angegebenen Pfade entsprechend anpassen.

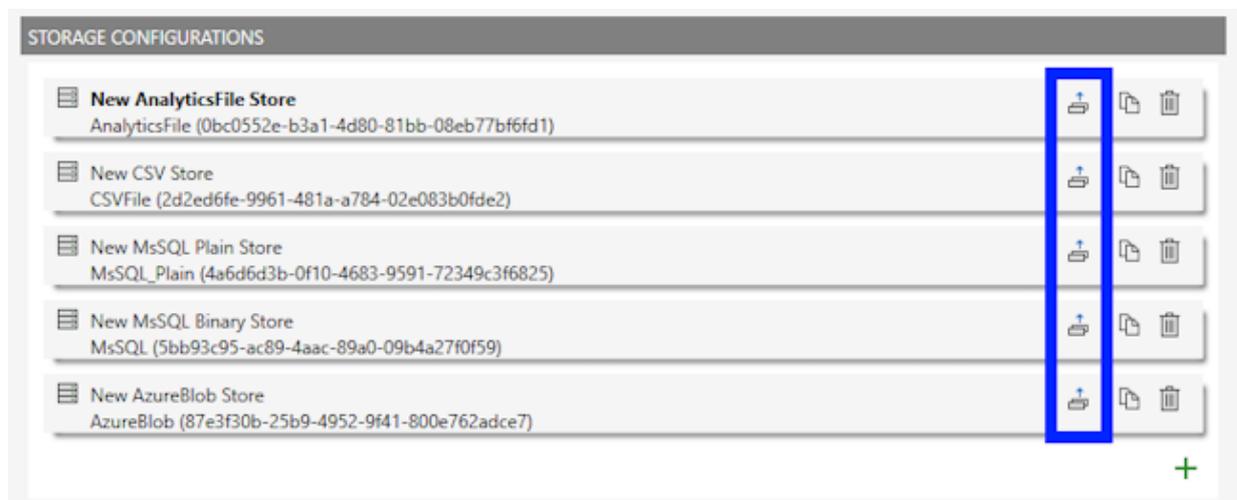
Sie können Aufnahmen aus dem Analytics Logger, die im Analytics File Format (*Analytics.tas*, *Analytics-<Date>.tay*) gespeichert sind, in den Storage Provider importieren. Generell können Sie Daten, die auch vom Storage Provider als Analytics File abgespeichert wurden, in andere Formate konvertieren. Analytics File ist dabei immer das Quellformat.

Führen Sie dazu folgende Schritte durch:

1. Legen Sie den Ordner mit den Analytics Dateien in Ihrem Storage Provider Speicherort ab, Standardmäßig hier: `C:\TwinCAT\Functions\TF3520-Analytics-StorageProvider\Storage\Analytics StorageProvider` (legen Sie die Ordner manuell an, falls diese nicht vorhanden sind)



2. Öffnen Sie die `TwinCAT.Analytics.StorageProvider.Configurator.exe`. Das Programm finden Sie unter dem Pfad `C:\TwinCAT\Functions\TF3520-Analytics-StorageProvider\Tools`
3. Wählen Sie den Storage aus, in den das Analytics File importiert werden soll und drücken Sie den Button **DataImport**.



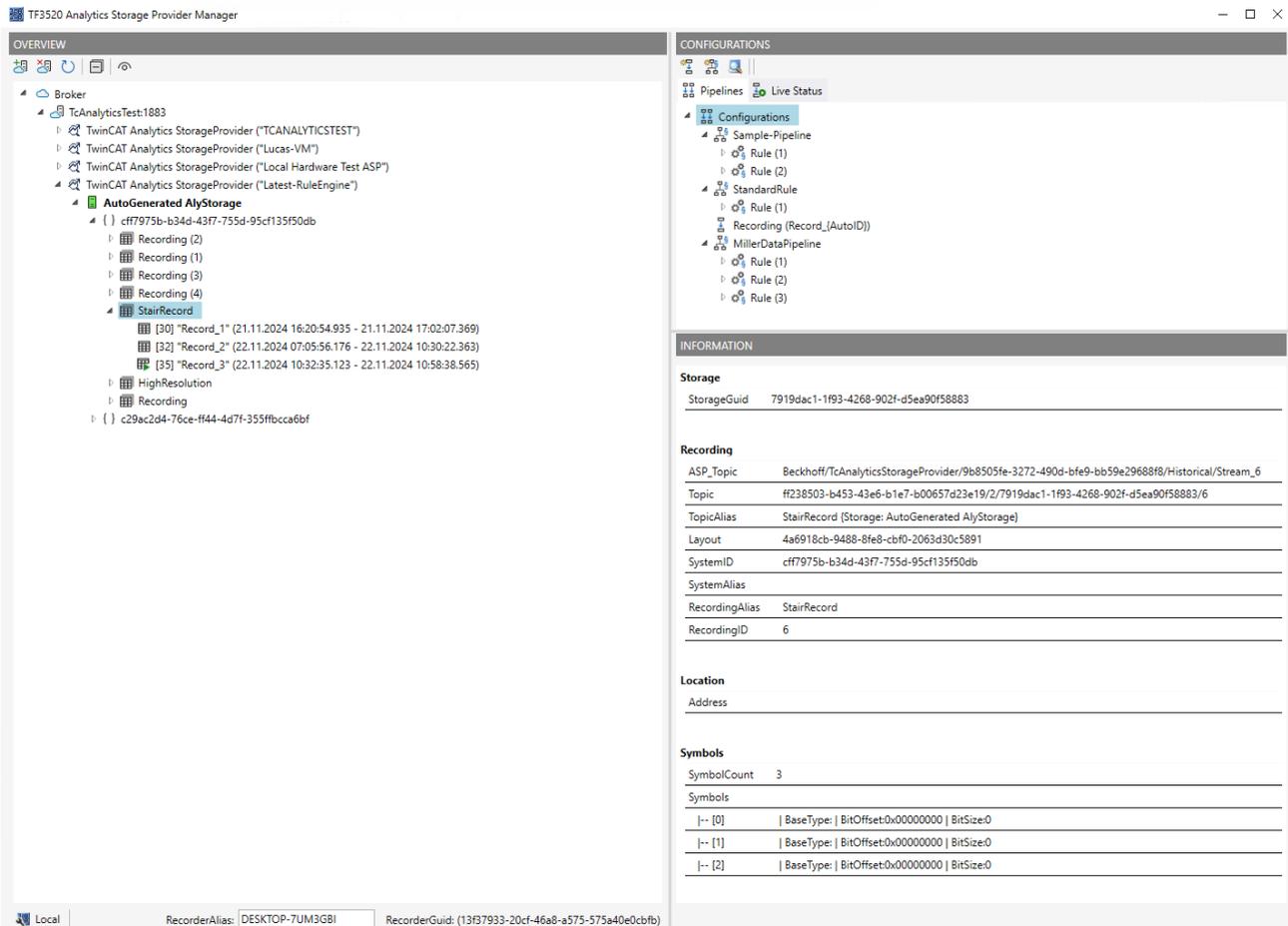
4. Wählen Sie dann im DataImport Dialog den Pfad zum Analytics File aus und geben alle weiteren bekannten Parameter zu dieser Aufnahme an.

5. Drücken Sie **OK** und der Datenimport beginnt.

- ⇒ Nun können Sie Ihre importierten Daten im [TwinCAT Target Browser \[► 99\]](#) sehen.
- ⇒ Möglicherweise müssen Sie kurz warten bzw. Ihren Storage Provider einmal neu starten.

6.2 Manager

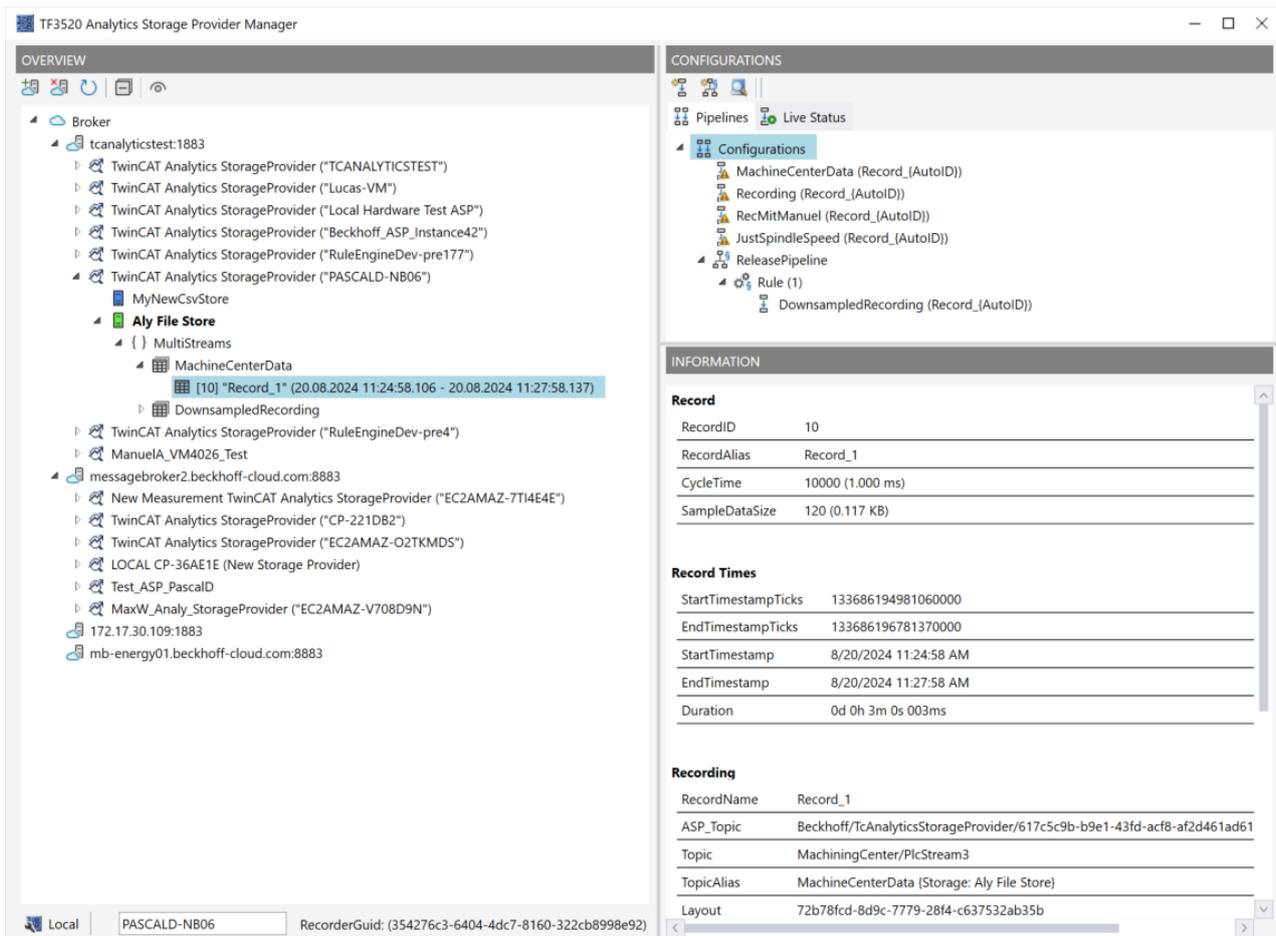
Im TwinCAT Analytics Storage Provider Manager können Sie Ihre Daten Pipelines definieren. Eine Pipeline definiert sich dabei so, dass Sie die Daten Quellen (Data Message Broker) und die dort vorhandenen Symbole auswählen können. Die Daten optional durch Rules vorverarbeiten und sie letztendlich in einem oder mehreren Stores ablegen können. Die Stores müssen dafür im [Configurator \[► 47\]](#) erstellt worden sein.



6.2.1 Manager („Recorder“)

Der Analytics Storage Provider Recorder ist Teil der Analytics Engineering-Setups. Daher finden Sie den Recorder in der Installation der TwinCAT Analytics Workbench und des TwinCAT Analytics Service Tools.

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 Provider und historisierten Daten werden übersichtlich dargestellt.



Symbolleiste Manager Fenster („OVERVIEW“)

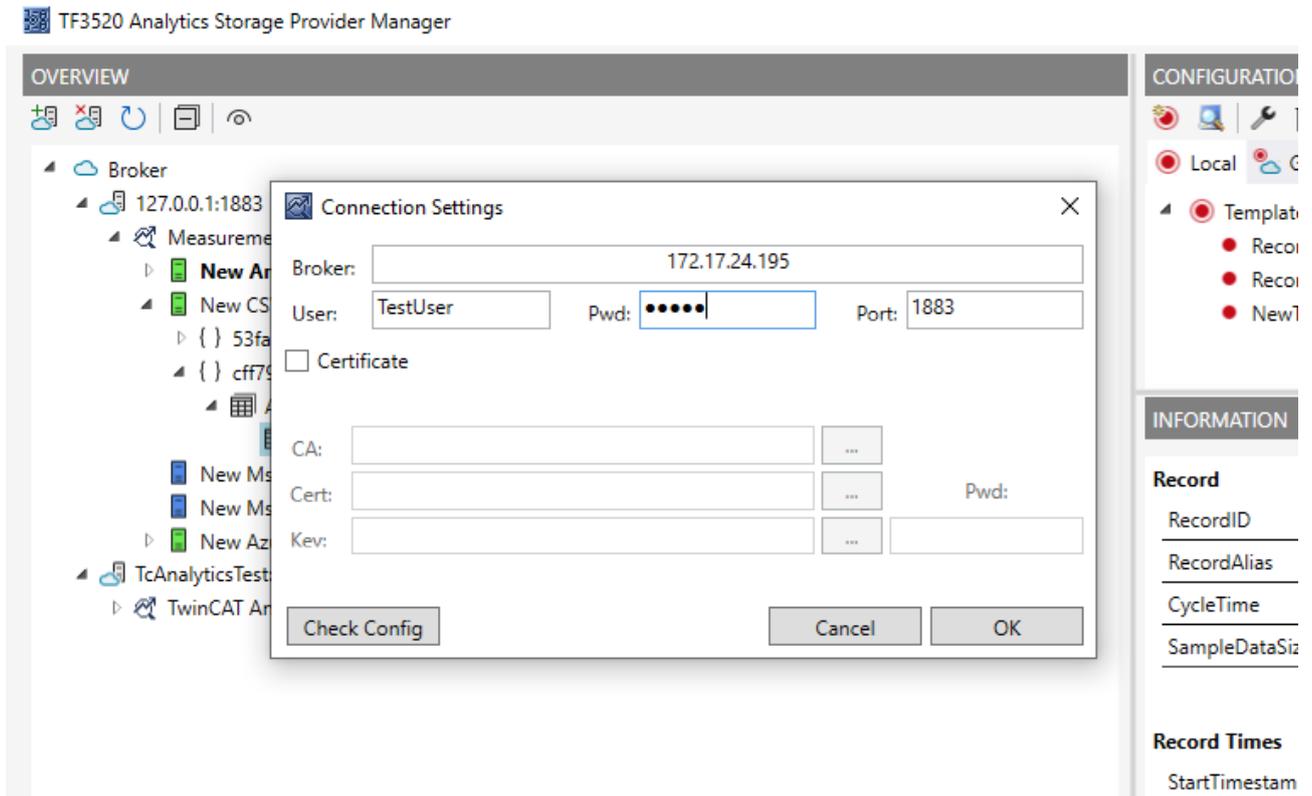


1	Neuen Broker hinzufügen
2	Ausgewählten Broker entfernen
3	Anzeige aktualisieren
4	Alle Knoten einklappen
5	Ansicht Wechsel zwischen Dark/Light Modus

Funktion Manager Fenster („OVERVIEW“)

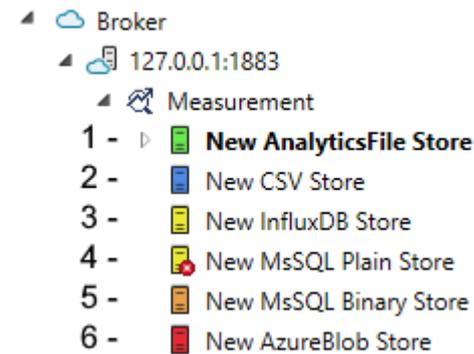
Vergeben Sie zunächst einen „RecorderAlias“. Dieser hilft die gestarteten Aufnahmen zu gruppieren und seine selbst gestarteten wieder zu finden.

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



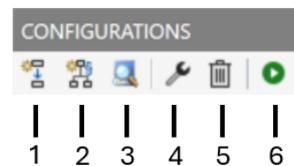
Sobald eine Verbindung zum Broker hergestellt werden konnte, werden alle darauf verbunden Analytics Storage Provider aufgelistet.

„Storage“ Status



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

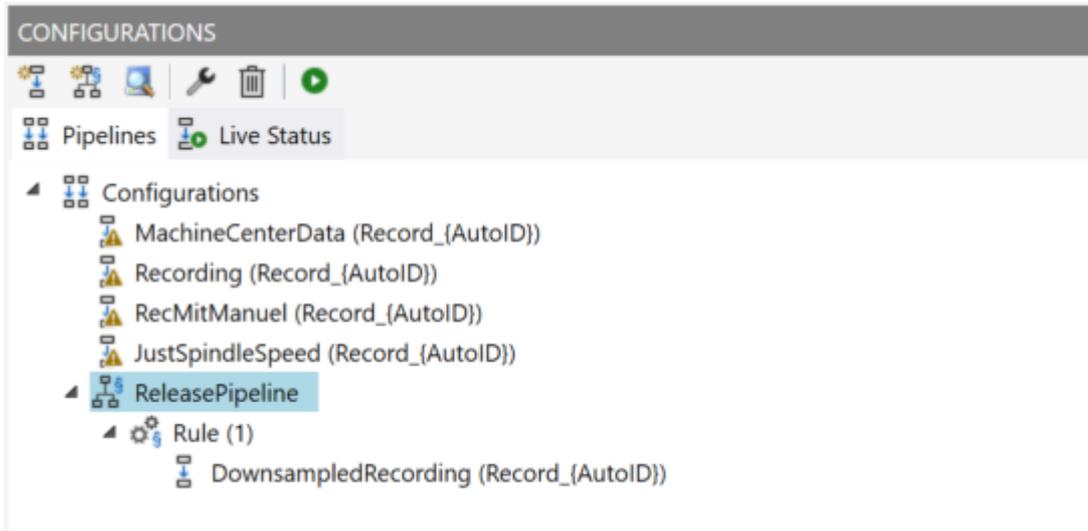
Symbolleiste Manager Fenster („CONFIGURATIONS“)



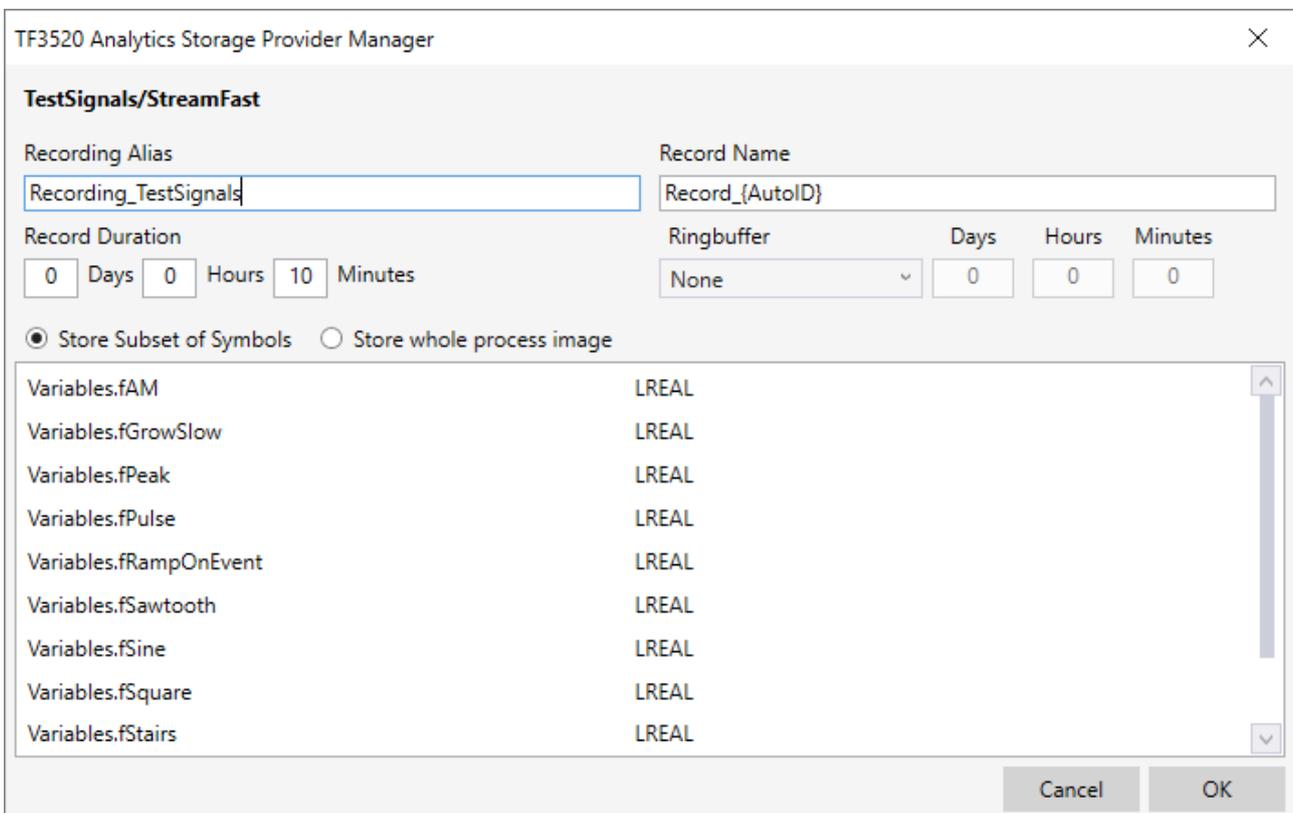
1	Eine neue Pipeline erzeugen
2	Eine neue Pipeline mit Rule Engine erzeugen
3	Target Browser öffnen für Beschaltung einfacher Pipelines
4	Eine ausgewählte Pipeline editieren
5	Eine ausgewählte Pipeline löschen
6	Eine ausgewählte Pipeline starten

Funktion Manager Fenster („CONFIGURATIONS“)

Das Fenster teilt sich in zwei Tabs auf. Pipelines und Live Status. Unter Pipelines finden Sie die Konfigurationen Ihrer Pipelines. Von hier können Sie neue Pipelines definieren, bestehende editieren, löschen oder starten.



Um eine neue einfache Pipeline zu erzeugen, drücken Sie den Button **Neue Pipeline erzeugen**. Folgender Dialog öffnet sich.



Sie können nun die Symbole, die aufgezeichnet werden sollen, vom Target Browser per Drag-and-Drop in den Dialog ziehen. Vergeben Sie zusätzlich ein Recording Alias und einen Record Name.

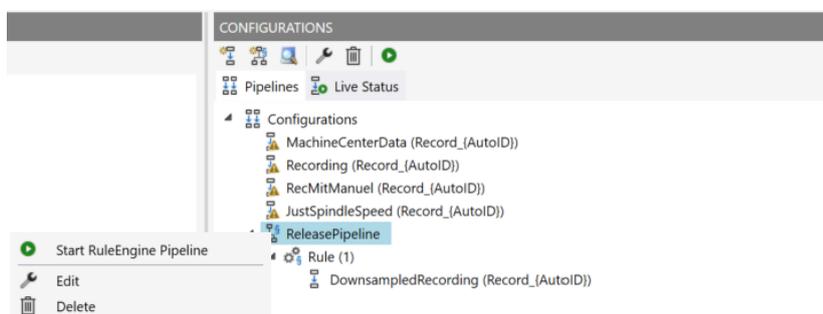
Für den Record Name stehen verschiedene Platzhalter zur Verfügung:

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

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

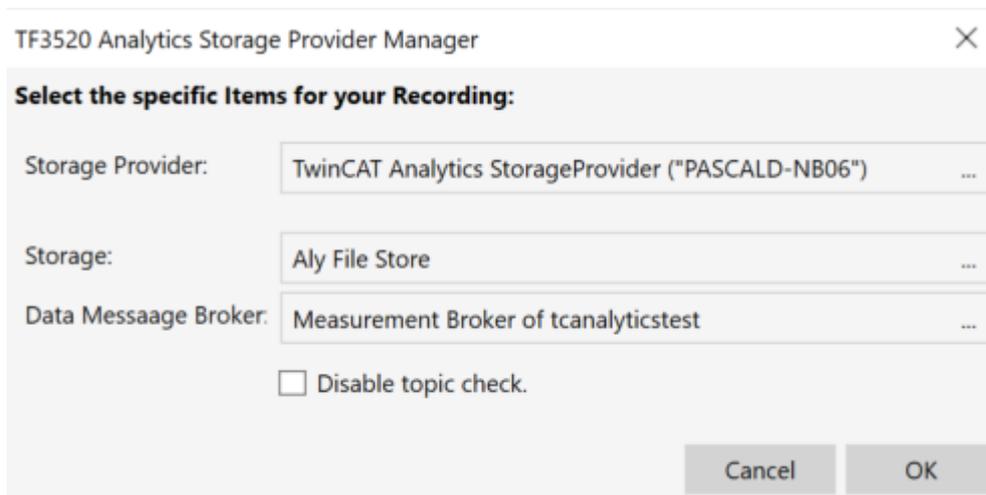
Mit **OK** werden die Eingaben bestätigt und eine neue lokale Aufnahmedefinition wird erzeugt.

Es ist nun möglich diese Definition direkt über die Toolbar oder das Kontextmenü zu starten.



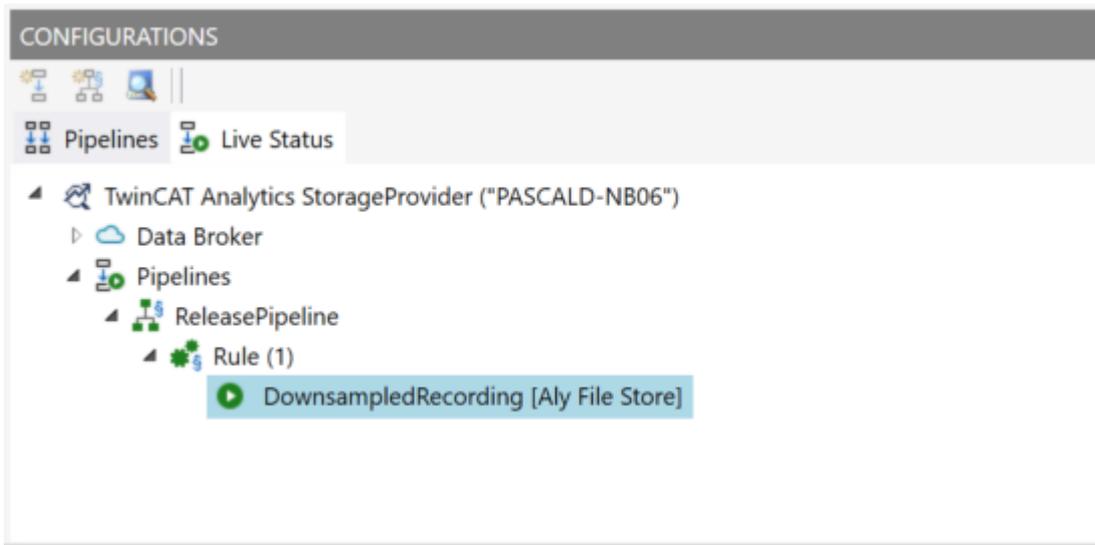
Es besteht aber auch die Möglichkeit die Definition global zugänglich zu machen. Dies kann über das Kontextmenü mit dem Eintrag **Publish Recording** gemacht werden.

Danach öffnet sich folgender Dialog:



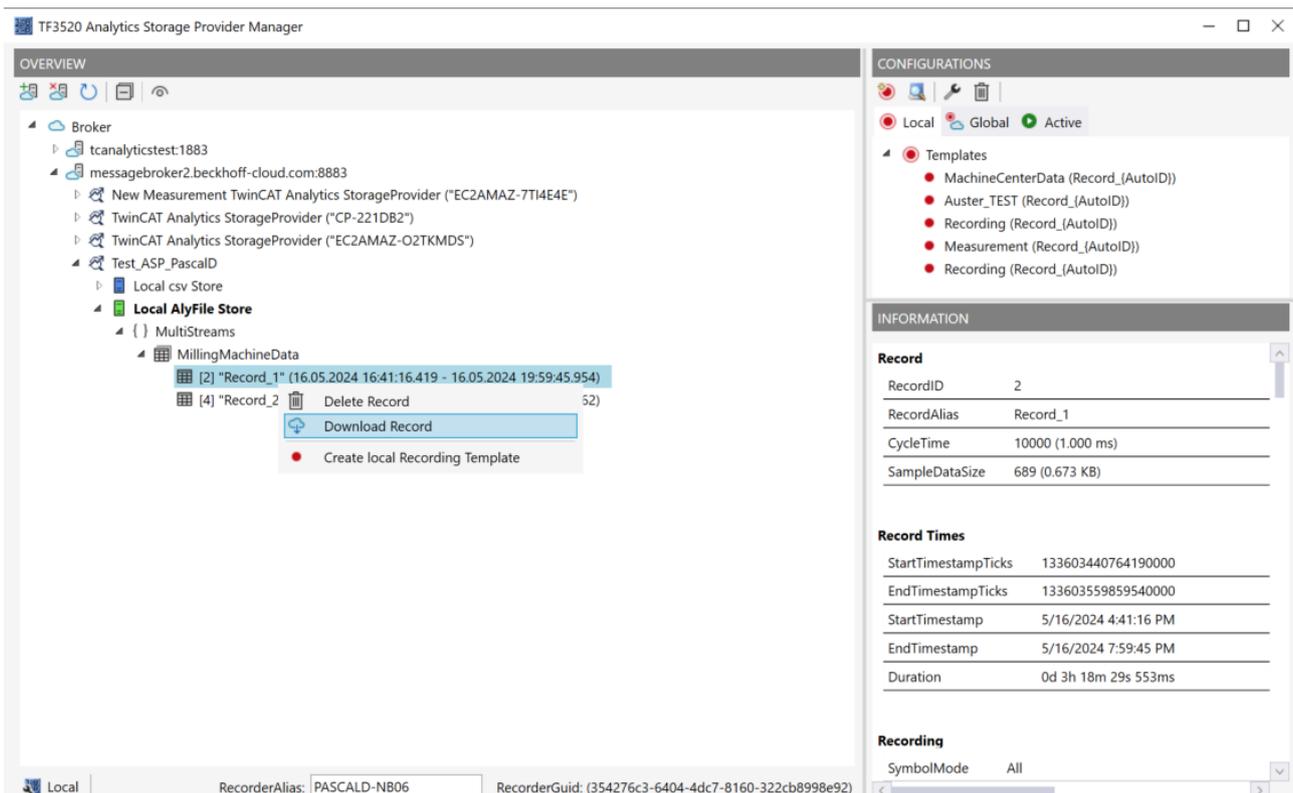
Hier kann nun der gewünschte Analytics Storage Provider ausgewählt werden, über den die Definition veröffentlicht werden soll. Zusätzlich wird der Definition ein Storage und ein Data Broker des ausgewählten Analytics Storage Providers zugewiesen. Nach der Auswahl wird die Aufnahmedefinition mit **OK**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

Nach dem Start einer Pipeline springt die Ansicht automatisch in den zweiten Tab, den Live Status.



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.

Des Weiteren steht Ihnen eine Download Funktion im Manager zur Verfügung, um Daten auf Ihr lokales Engineering herunterzuladen. Suchen Sie sich den entsprechenden Record aus und starten den Download über das Kontextmenü. Unabhängig davon in welchem Store die Daten liegen, werden sie grundsätzlich als Analytics File auf dem lokalen Engineering System abgespeichert.



In Kombination mit der [Datenimport-Funktion](#) [► 57] im Storage Konfigurator fungiert das Analytics File auch als praktisches Austauschformat. Mit beiden Funktionen können die Daten beispielsweise von einer CSV-Datei in eine InfluxDB2 verschoben werden oder von einer Microsoft SQL® in eine CSV-Datei, usw.

6.2.2 Datenmanagement mit Rule Engine

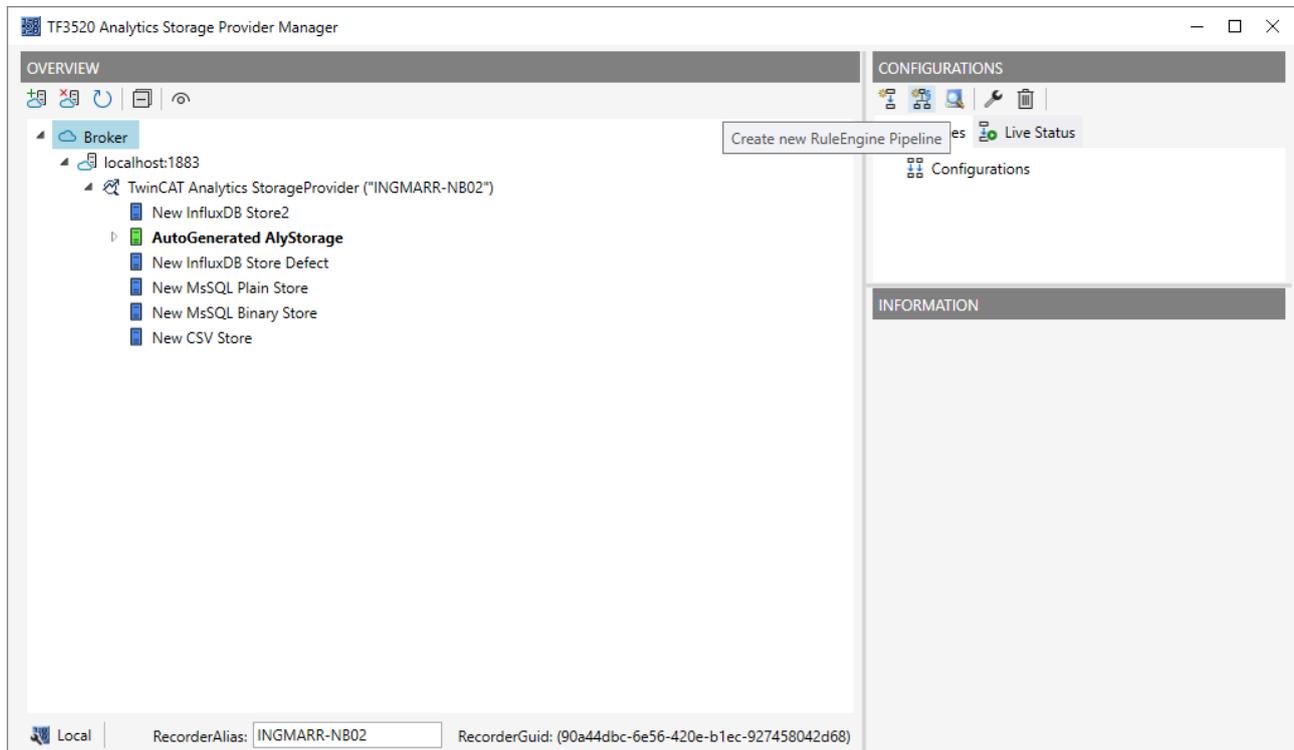
Mit Rule Engine-Pipelines können unterschiedlichste Speicherszenarien automatisiert abgebildet werden. Daten können aus mehreren MQTT-Datenquellen erfasst, verarbeitet und in Recordings abgelegt werden. Zudem können Daten aus Recordings event- oder zeitgesteuert, sowie manuell an getriggert ausgelesen und weiterverarbeitet werden. Mögliche Verarbeitungsoperationen sind das Aggregieren, Filtern und Samplen der Datenquellen.



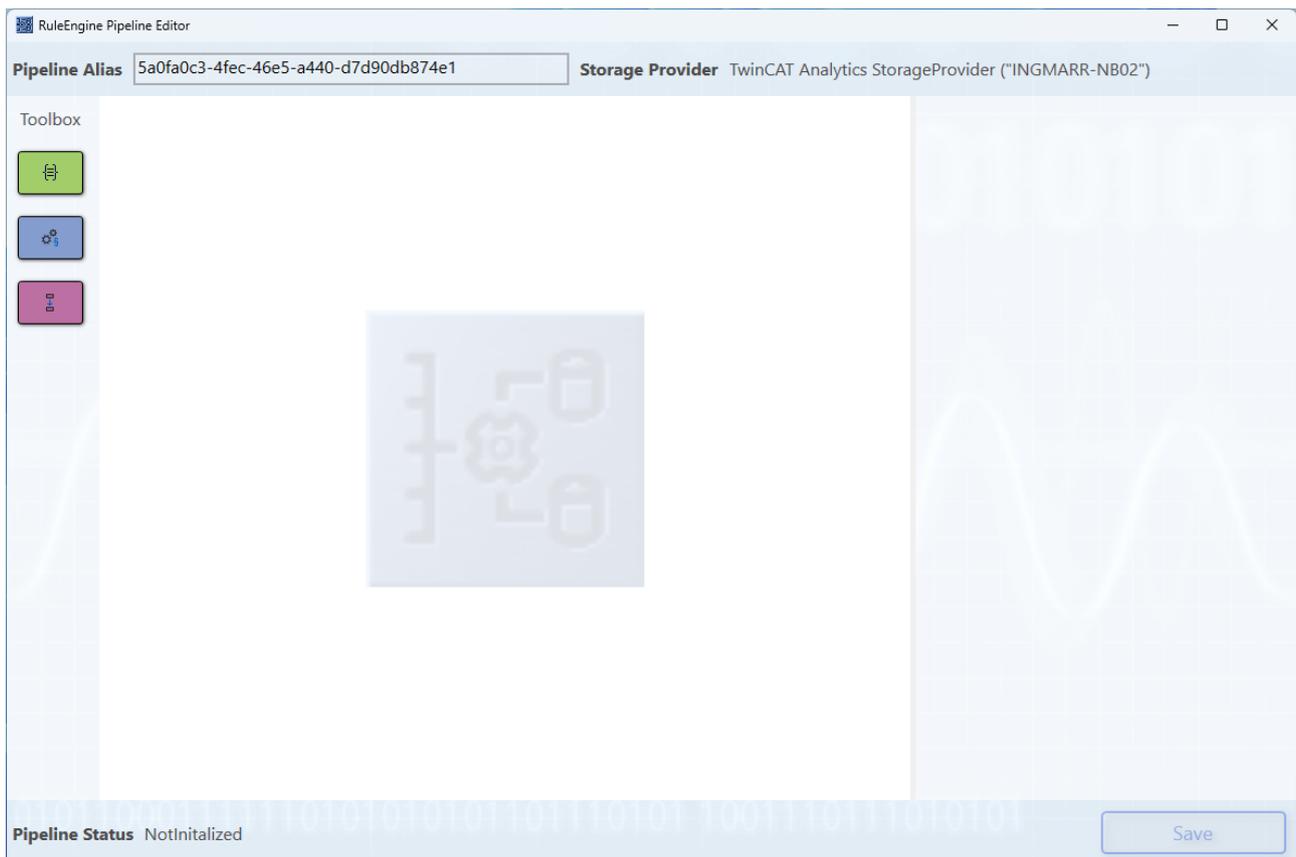
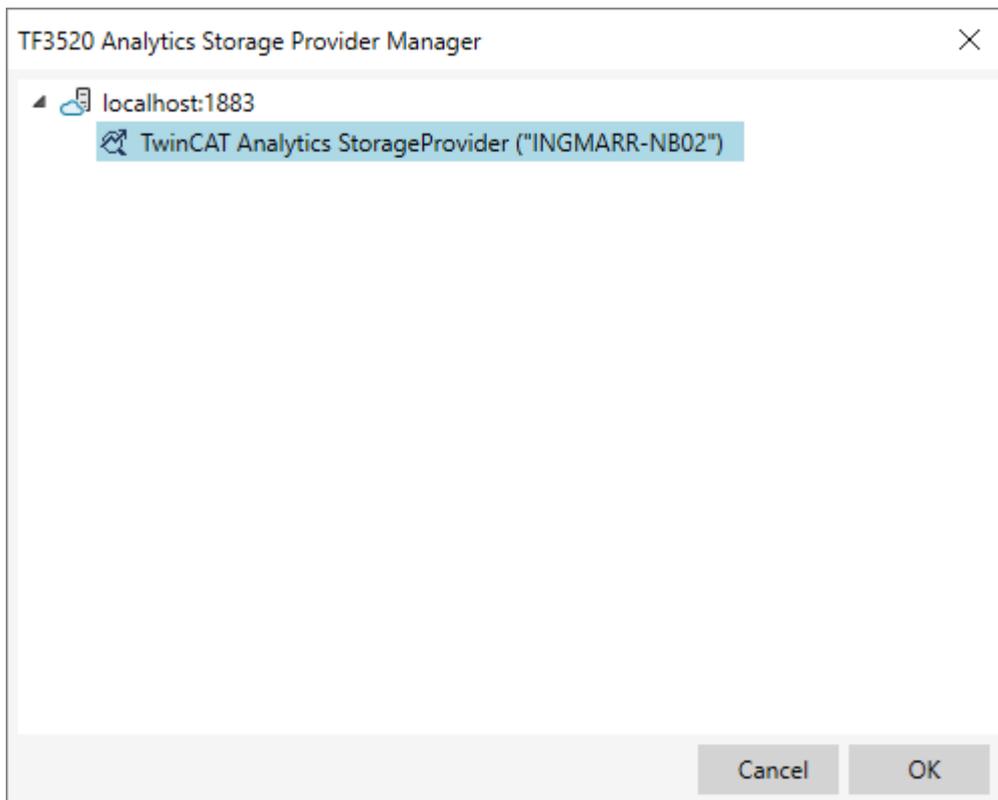
Die RuleEngine-Pipeline-Funktion steht ab Storage Provider Version 3.15. zur Verfügung.

6.2.2.1 Konfiguration

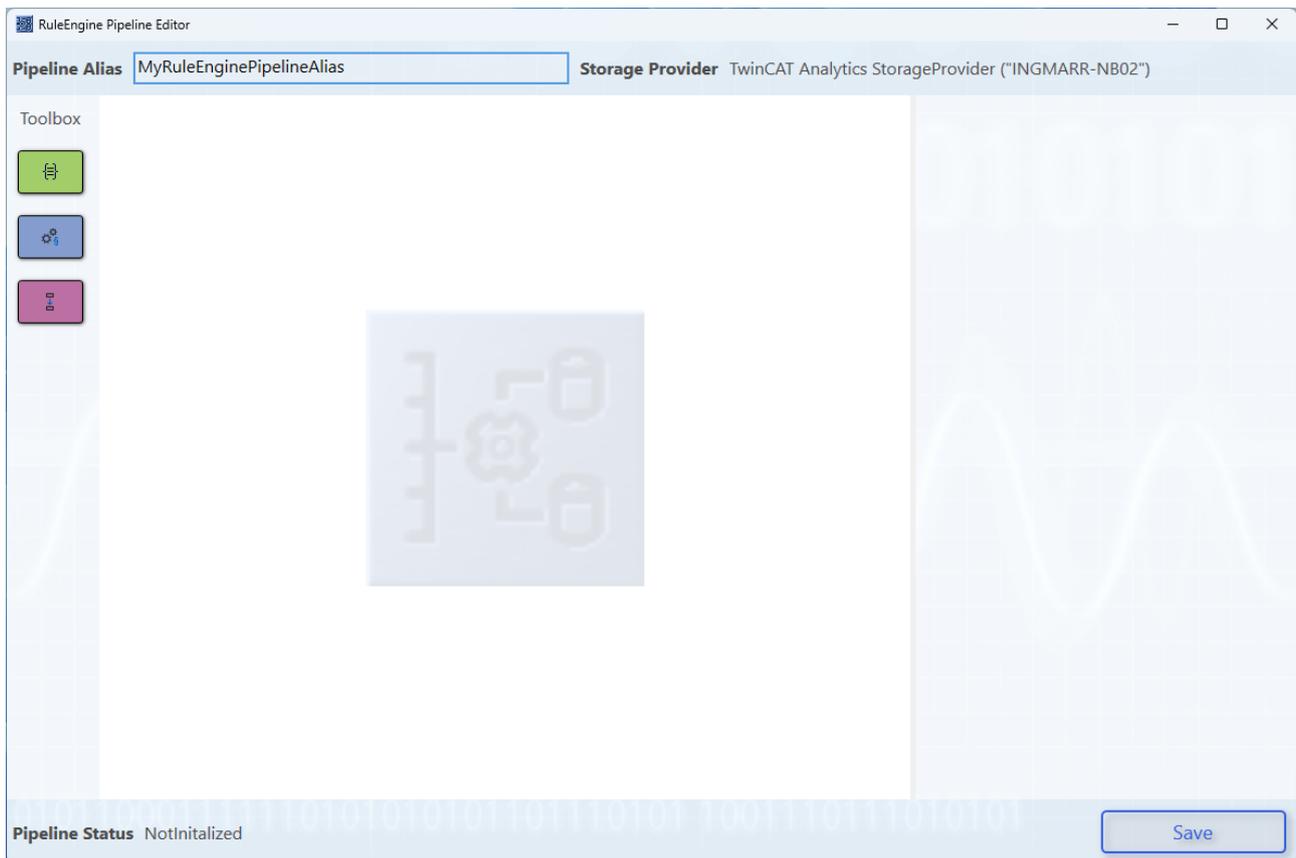
Die Konfiguration einer RuleEngine-Pipeline erfolgt grafisch im RuleEngine-Pipeline-Editor. Dieser kann über den Manager geöffnet werden.



Zunächst muss der gewünschte Storage Provider ausgewählt werden. Eine RuleEngine-Pipeline wird speziell für einen Storage Provider entwickelt. Dieser kann nicht im Nachgang geändert werden. Im RuleEngine-Pipeline-Editor wird der Name des gewählten Storage Providers oben rechts dargestellt.



Für eine RuleEngine-Pipeline kann ein Alias vergeben werden. Der Alias ist frei wählbar und kann über das Textfeld oben links im RuleEngine-Pipeline Editor bearbeitet werden.



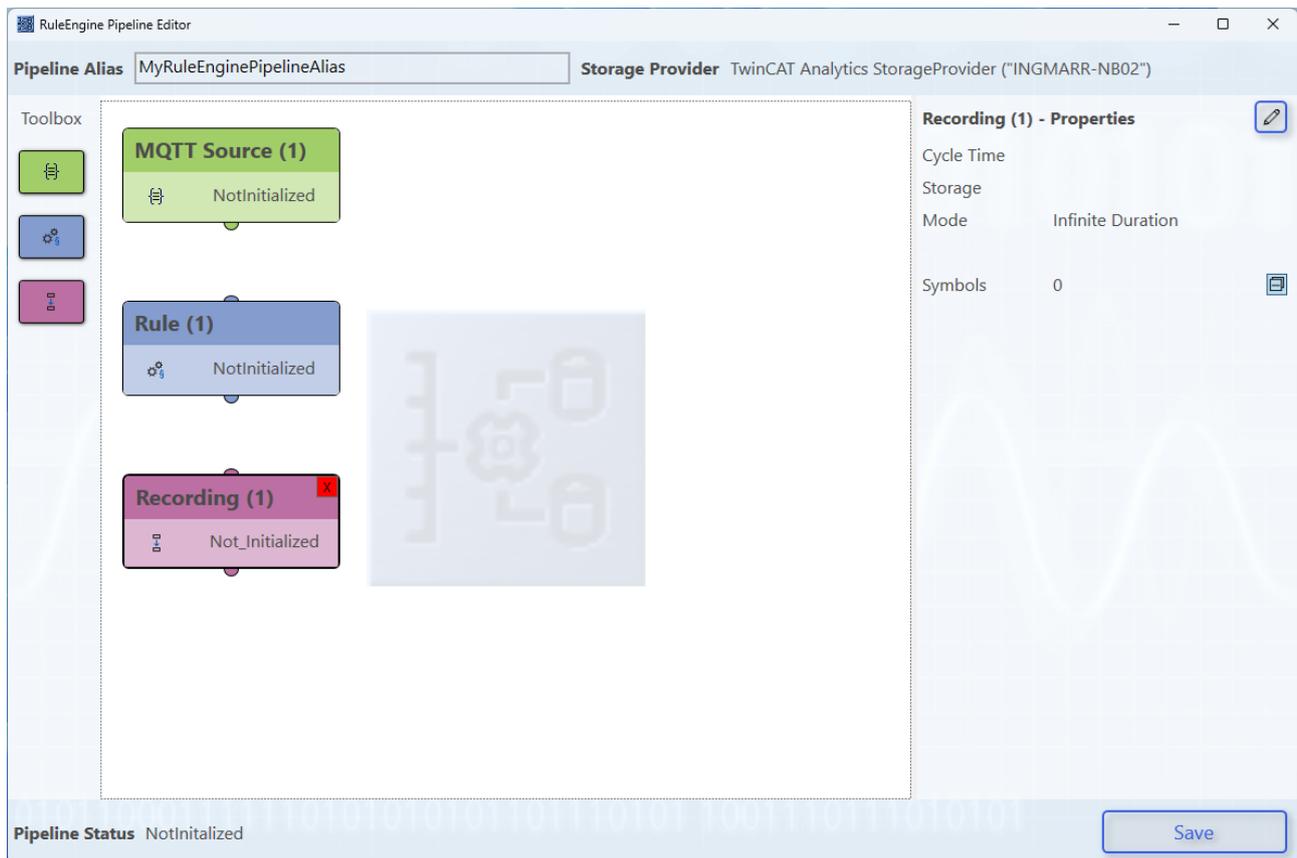
6.2.2.1.1 Hinzufügen von Elementen

Aus der Toolbox im RuleEngine-Pipeline-Editor (auf der linken Seite) können unterschiedliche Elemente in den Konfigurationsbereich (Mitte) gezogen werden. Folgende Elemente können verwendet werden:

MQTT-Source: Entspricht einer Datenquelle (bspw. TwinCAT Analytics Logger oder TwinCAT IoT Data Agent).

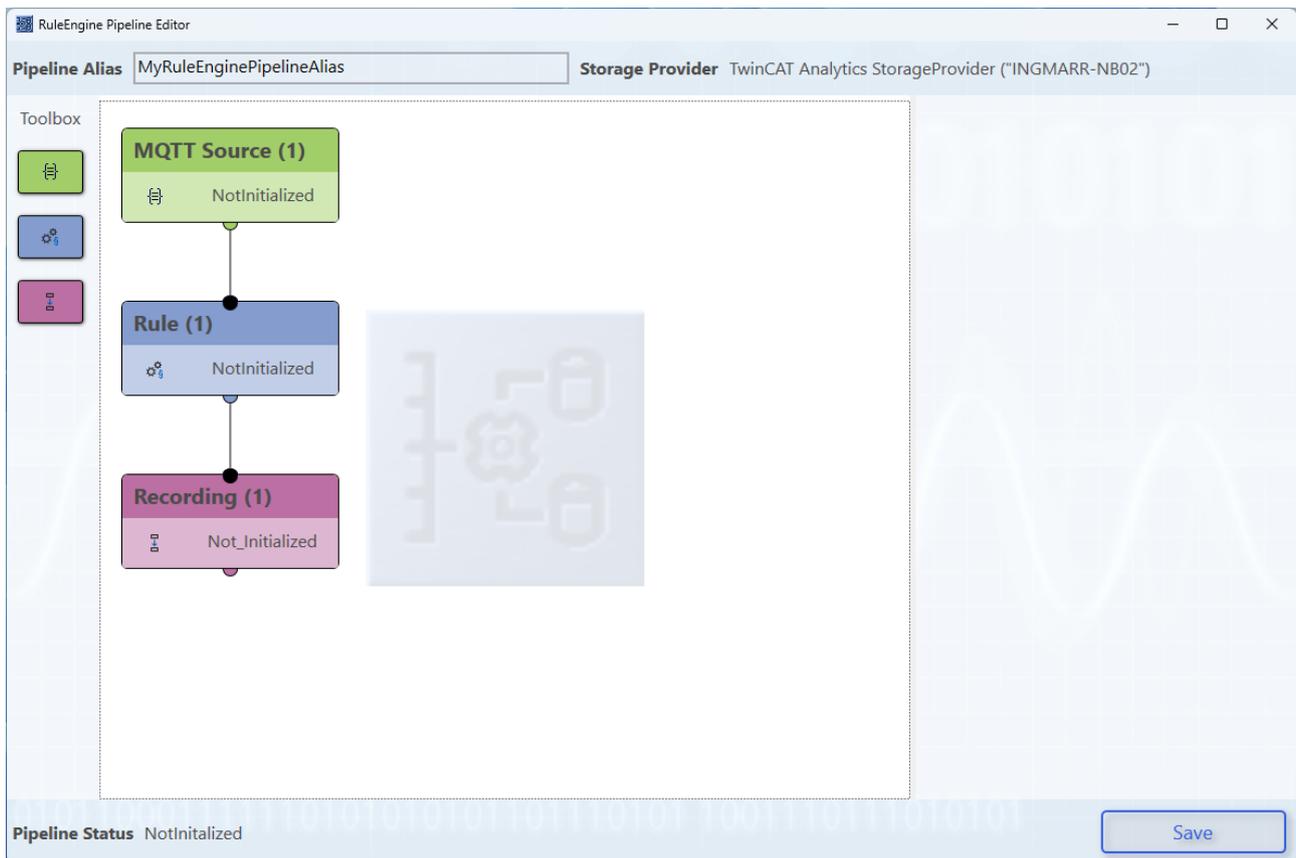
Rule: Mit einer Rule können Vorverarbeitungsschritte definiert werden.

Recording: Beschreibt eine Datenaufnahme (bisher bekanntes Recording).



6.2.2.1.2 Verknüpfen von Elementen

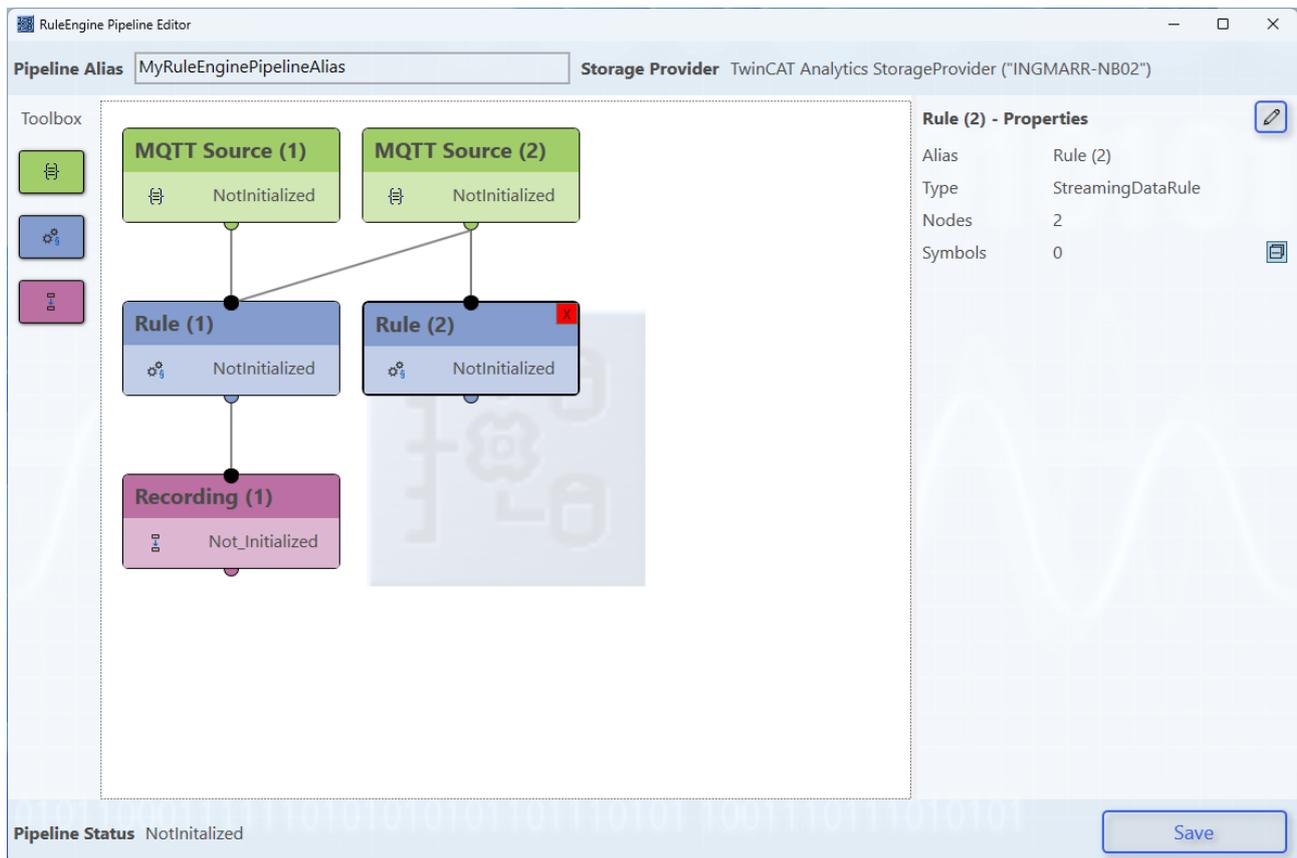
Die hinzugefügten Elemente können über Verbindungen miteinander verknüpft werden. Eine Verbindung kann über den Ausgang eines Elements erzeugt werden. Mit der Maus kann die Verbindung auf ein anderes Element gezogen werden und so eine Verknüpfung im Datenfluss hergestellt werden. Lässt man den Verbindner im freien Bereich los, wird automatisch ein nachfolgendes Element erstellt bzw. eine Auswahl über verfügbare Elementtypen angeboten.



Folgende Verbindungen sind möglich:

MQTT-Source – Rule:

Eine MQTT-Source kann in mehreren Rules verwendet werden. Zudem können mehrere MQTT-Sources in einer Rule verwendet werden, sofern diese mit ähnlichen Systemzeiten laufen. Ist dies nicht der Fall, erkennt das der Editor und gibt eine entsprechende Warnung aus.



Rule – Recording:

Für eine Rule können beliebig viele Recordings zum Abspeichern von Daten definiert werden. In ein Recording können jedoch nur Daten aus einer Rule reinfließen. Für ein Recording können beliebig viele Rules für die Weiterverarbeitung der gespeicherten Daten angelegt werden. Auch hier kann eine Rule nur Daten aus einem Recording entgegennehmen. Bei Recordings mit einem Ringspeicher können derzeit keine nachfolgenden Rules angelegt werden. Auch die Kombination von Daten aus einer MQTT-Source und einem Recording als Datenquellen für eine Rule ist nicht möglich.

The screenshot shows the RuleEngine Pipeline Editor interface. The main window displays a pipeline diagram with the following components and connections:

- MQTT Source (1)** (green box) is connected to **Rule (1)** (blue box).
- Rule (1)** is connected to **Recording (1)** (pink box) and **Recording (2)** (pink box).
- Recording (1)** is connected to **Rule (2)** (blue box) and **Rule (3)** (blue box).
- Recording (2)** is connected to **Rule (3)**.

The **Rule (3) - Properties** window on the right shows the following details:

Rule (3) - Properties	
Alias	Rule (3)
Type	BatchDataRule
Nodes	2
Symbols	0

Below the properties window, the **Rule Trigger** section shows:

Rule Trigger	
Trigger	0

The **Pipeline Status** at the bottom left is **NotInitialized**. A **Save** button is located at the bottom right.

MQTT-Source – Recording:

Diese Verbindung ist nicht möglich. Es muss immer eine Rule als Zwischenverbindung eingefügt werden.

6.2.2.1.3 Einsehen von Eigenschaften der Elemente

Über das Properties-Fenster auf der rechten Seite des RuleEngine-Pipeline Editors können die Eigenschaften des selektierten Elements eingesehen werden. Die Eigenschaften variieren je nach Typ.

MQTT-Source

- **Broker:** Message Broker der Datenquelle.
- **Stream:** Topic der Datenquell
- **SystemID:** System-ID der Datenquelle
- **Cycle Time:** Zykluszeit der Datenquelle
- **Symbols:** Anzahl sowie Namen der ausgewählten Symbole, welche an der Datenquelle ausgewählt wurden.

Rule

- **Alias:** Alias-Name der Rule
- **Type:** Typ der Regel. Derzeit gibt es StreamingDataRules und BatchDataRules. StreamingDataRules beziehen sich auf gestreamte Daten und laufen kontinuierlich mit, um die gestreamten Daten zu verarbeiten. BatchDataRules beziehen sich auf bereits gespeicherte Daten. Diese laufen nicht kontinuierlich mit, sondern werden über Trigger aufgestartet. Sie verarbeiten eine gewisse Menge an gespeicherten Daten und fahren anschließend wieder herunter.
- **Nodes:** Anzahl der Knoten in einer Rule.
- **Symbols:** Anzahl sowie Namen der Symbole, welche aus der Rule herausgegeben werden.
- **RuleTrigger** (nur bei BatchDataRules): Anzahl der konfigurierten Trigger.

The screenshot displays the RuleEngine Pipeline Editor interface. The main workspace shows a vertical pipeline with three nodes: a green 'Demodata/PlcStr...' node at the top, a blue 'Rule (1)' node in the middle, and a pink 'Recording (1)' node at the bottom. All nodes are marked as 'NotInitialized'. The 'Pipeline Alias' is 'MyRuleEnginePipelineAlias' and the 'Storage Provider' is 'TwinCAT Analytics StorageProvider ("INGMARR-NB02")'. The 'Pipeline Status' at the bottom left is 'NotInitialized'. The right-hand 'Rule (1) - Properties' panel shows the following details:

Property	Value
Alias	Rule (1)
Type	StreamingDataRule
Nodes	4
Symbols	16

The 'Cyclic Data' section lists the following symbols:

- MAIN.Druckverlust_Avg (LREAL)
- MAIN.Druckverlust_Max (LREAL)
- MAIN.Druckverlust_Min (LREAL)
- MAIN.Gewicht_Avg (LREAL)
- MAIN.Gewicht_Max (LREAL)
- MAIN.Gewicht_Min (LREAL)
- MAIN.Kuehlmitteltemperatur_Avg
- MAIN.Kuehlmitteltemperatur_Max
- MAIN.Kuehlmitteltemperatur_Min

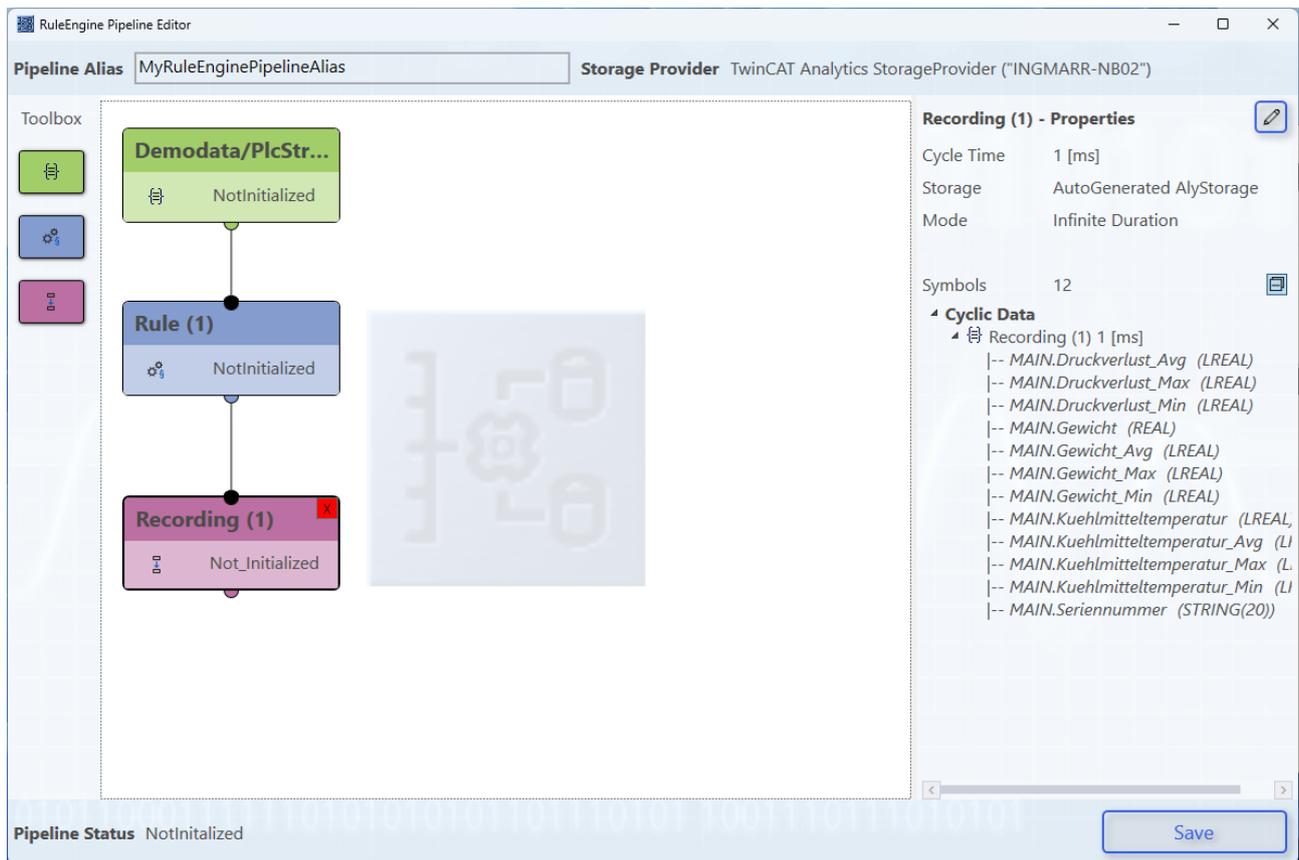
The 'FilterData' section lists:

- Demodata/PlcStream1 1 [ms]

A 'Save' button is located at the bottom right of the interface.

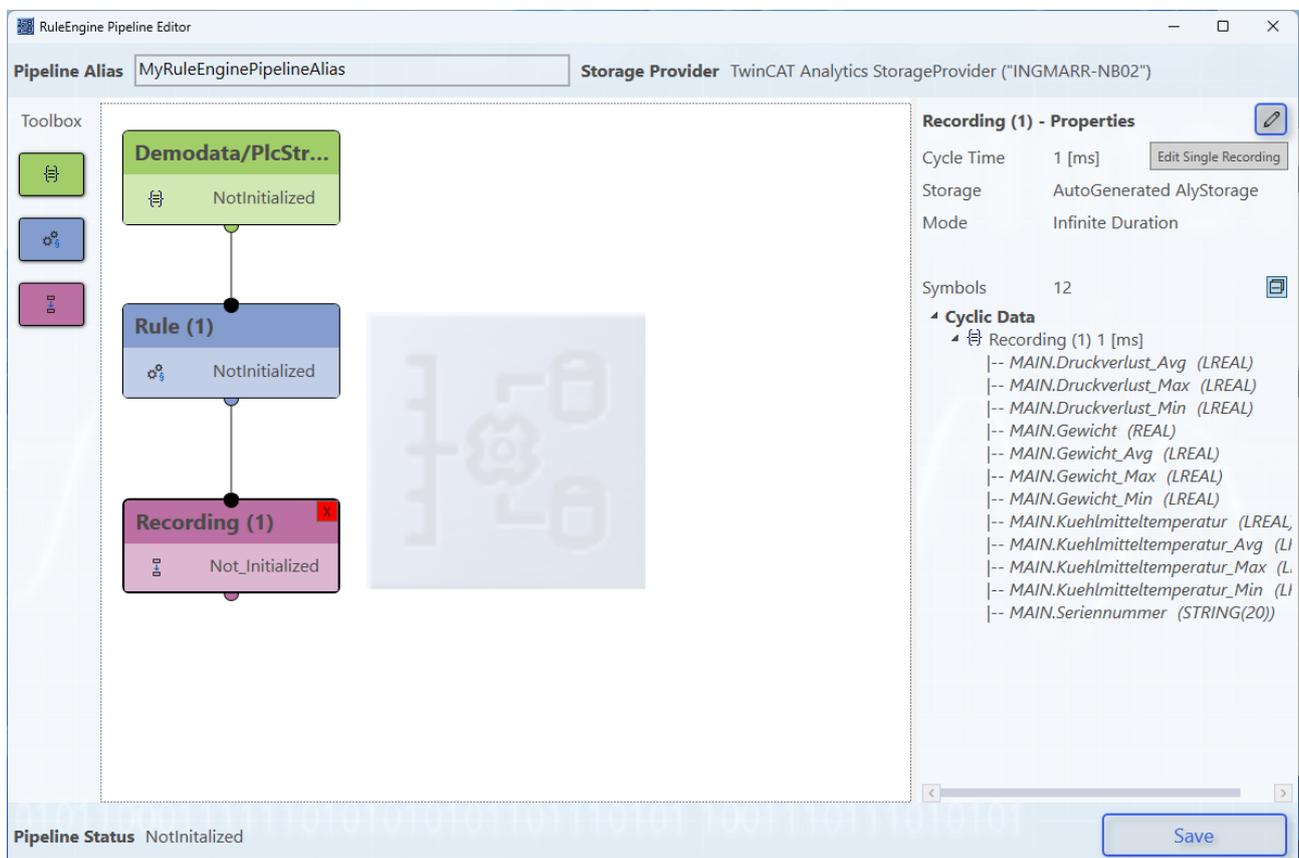
Recording

- **CycleTime:** Zykluszeit der Datenaufnahme
- **Storage:** Name des Datenspeichers
- **Mode:** Aufnahmemodus (Unbegrenzt oder Ringspeicher)
- **Symbols:** Anzahl sowie Namen der ausgewählten Symbole



6.2.2.1.4 Bearbeiten von Elementen

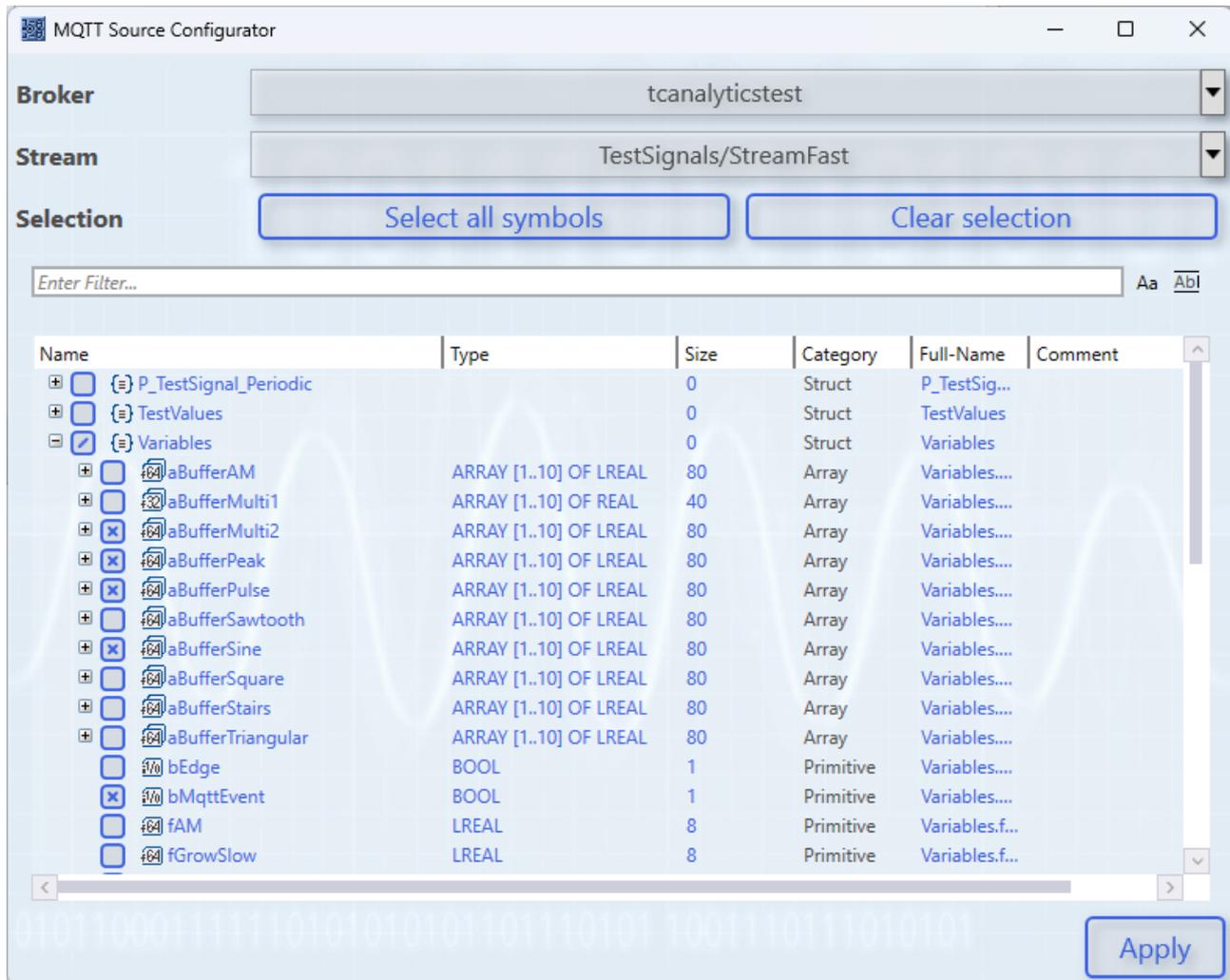
Die Bearbeitung erfolgt in einem separaten Fenster. Dieses kann über einen Doppelklick auf das Element oder das Bearbeitungssymbol in den Properties geöffnet werden.



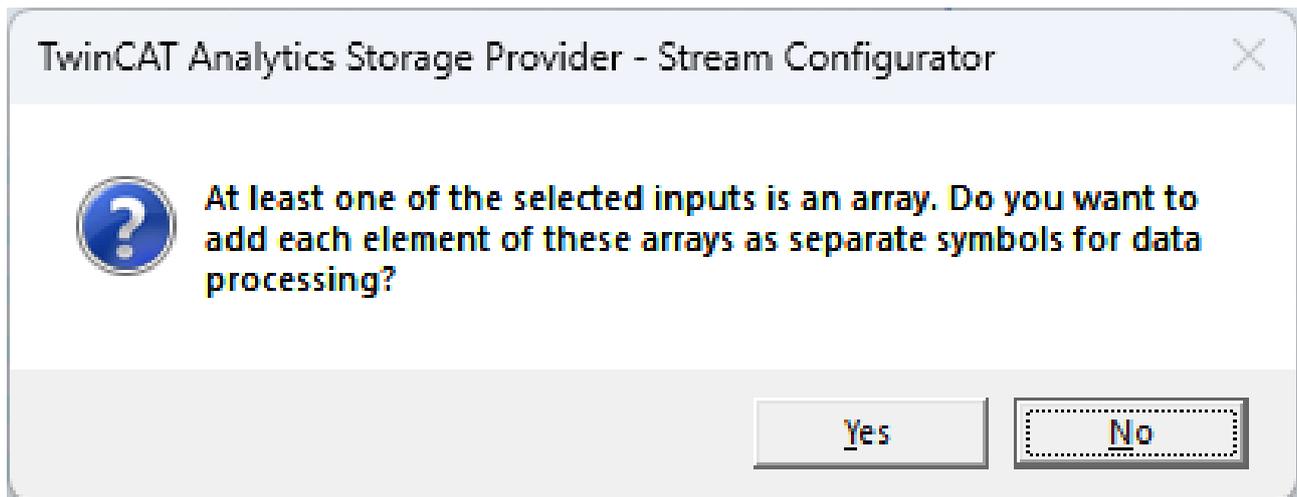
6.2.2.1.4.1 Bearbeiten von MQTT-Sources

Die Bearbeitung einer MQTT-Source erfolgt im MQTT-Source-Konfigurator. Zunächst muss unter dem Punkt Broker ein Message Broker ausgewählt werden. Ausgewählt werden können alle Data Broker des für die RuleEngine-Pipeline verwendeten Storage Providers, welche ebenfalls im Manager hinzugefügt wurden. Anschließend kann ein MQTT-Stream (Stream) basierend auf dessen Topic ausgewählt werden.

Sobald ein Stream hinzugefügt wurde, werden die verfügbaren Symbole des Streams angezeigt. Mittels der Checkboxes lassen sich einzelne Symbole selektieren. Über die Buttons **Select all symbols** und **Clear selection** können alle dargestellten Symbole selektiert bzw. deselektiert werden.



Befinden sich unter den ausgewählten Symbolen Arrays oder Strukturen wird beim Speichern der Konfiguration in einem Benutzerdialog abgefragt, ob die Subsymbole als einzelne Symbole hinzugefügt werden. Wenn diese hinzugefügt werden, können die Subsymbole einzeln in Rules oder Recordings für die Verarbeitung und Speicherung verwendet werden.



6.2.2.1.4.2 Bearbeiten von Rules

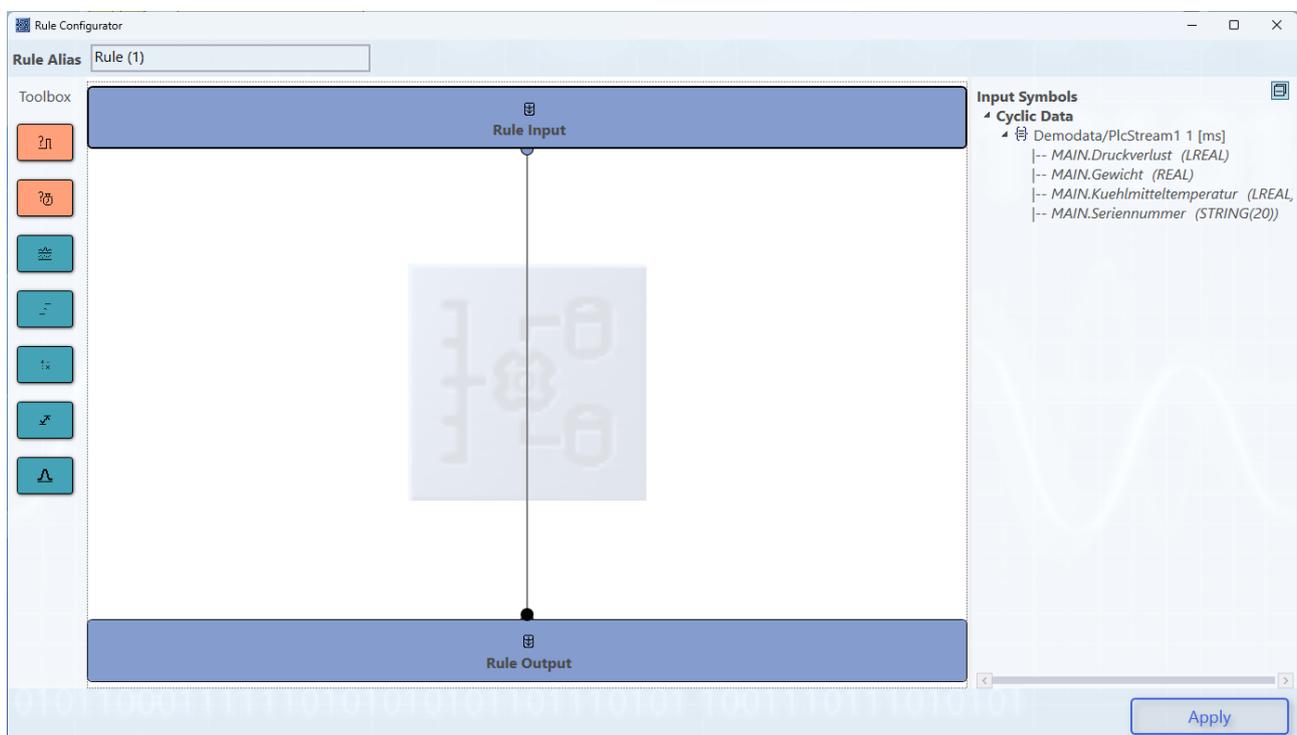
Die Bearbeitung einer Rule erfolgt im Rule-Konfigurator. Für eine Rule kann ein Alias-Name oben links vergeben werden. Eine Rule besteht aus folgenden Elementen:

Rule-Input:

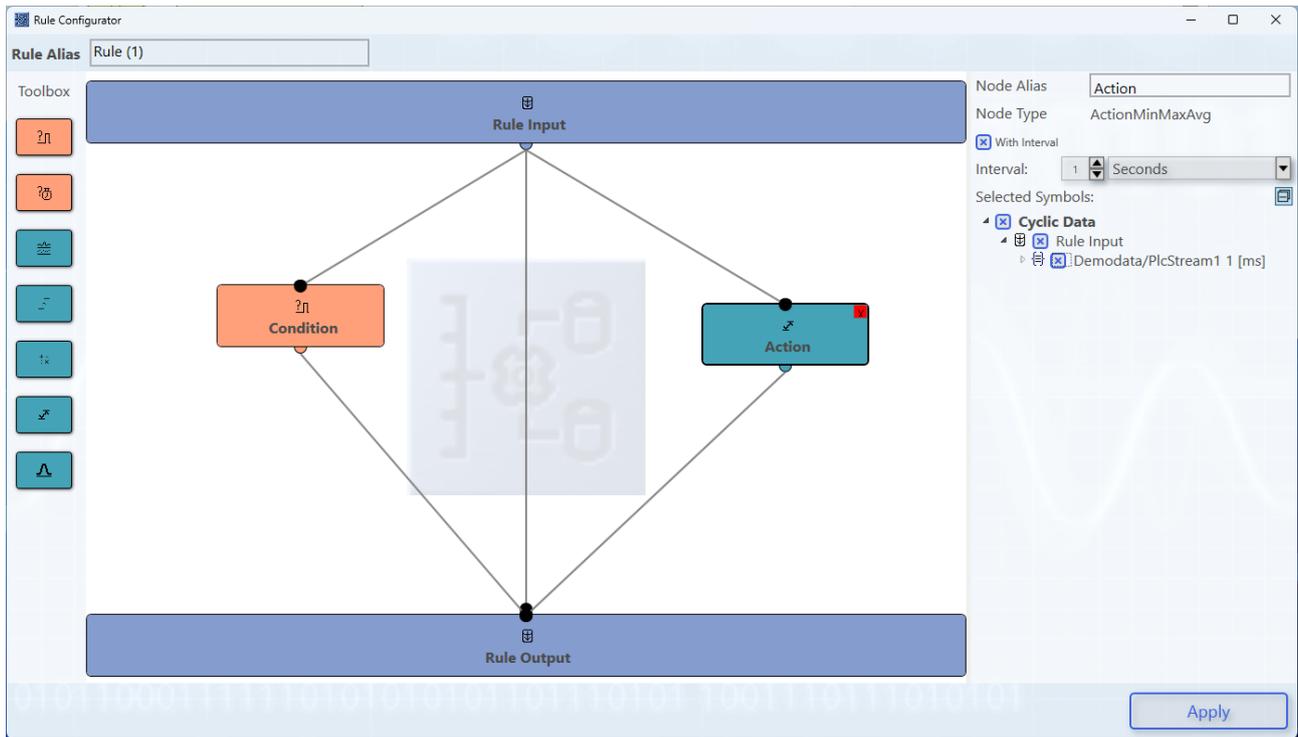
Entspricht dem Startpunkt innerhalb einer Rule. Hier liegen alle zu der Rule hinzugefügten Symbole an. Für BatchDataRules kann hier der Zeitbereich ausgewählt werden, für den beim Triggern der Rule die Daten geladen werden.

Rule-Output:

Entspricht dem Ausgang der Rule. Alle Symbole der Bausteine, welche mit dem Rule-Output verbunden sind, können für nachfolgende Recordings ausgewählt werden.

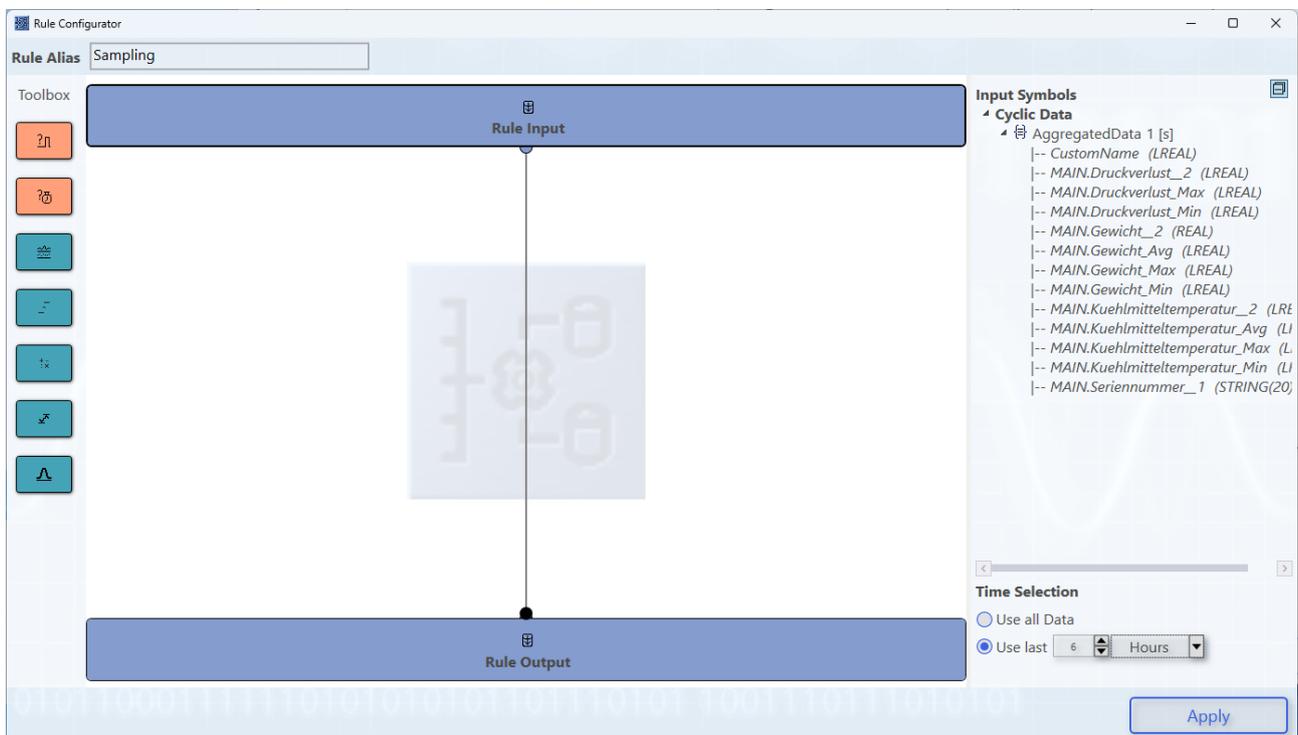


Über die Toolbox (links) können weitere Bausteine zur Verarbeitung hinzugefügt werden. Hierbei wird zwischen Condition-Bausteinen und Action-Bausteinen unterschieden. Condition-Bausteine dienen als Filter und ermöglichen es zeitbasiert oder datenbasiert die eingehenden Daten zu filtern. Action-Bausteine bieten die Möglichkeit zur Aggregation der Daten (wie bspw. die Berechnung von Min/Max/Avg oder weiteren Operationen). Alle Bausteine können untereinander mit Verbindungen verknüpft werden.



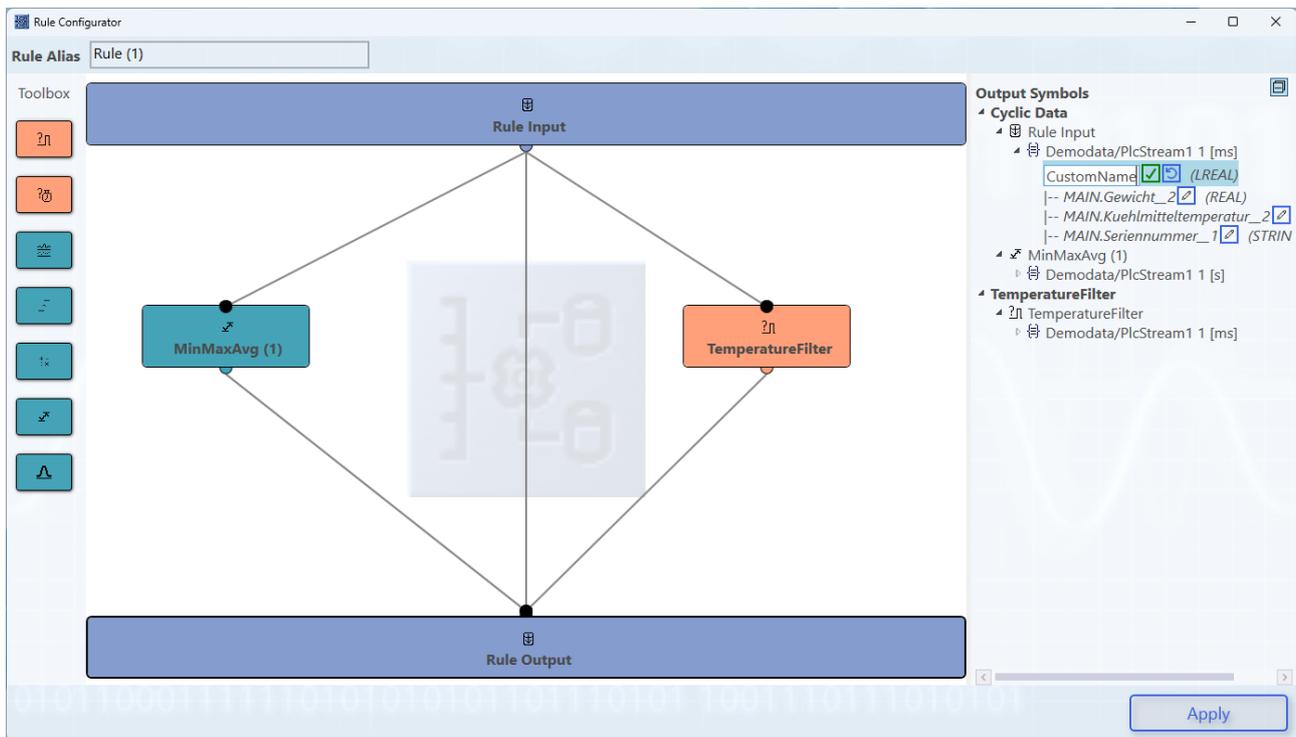
6.2.2.1.4.2.1 Input

Entspricht dem Startpunkt innerhalb einer Rule. Hier liegen alle zu der Rule hinzugefügten Symbole an. Für BatchDataRules kann hier der Zeitbereich ausgewählt werden, für den beim Triggern der Rule die Daten geladen werden.



6.2.2.1.4.2.2 Output

Entspricht dem Ausgang der Rule. Alle Symbole, der Bausteine, welche mit dem Rule-Output verbunden sind, können für nachfolgende Recordings ausgewählt werden. Am Output-Node können Variablen umbenannt werden. Hierfür kann man auf das Edit-Icon hinter der entsprechenden Variable klicken.



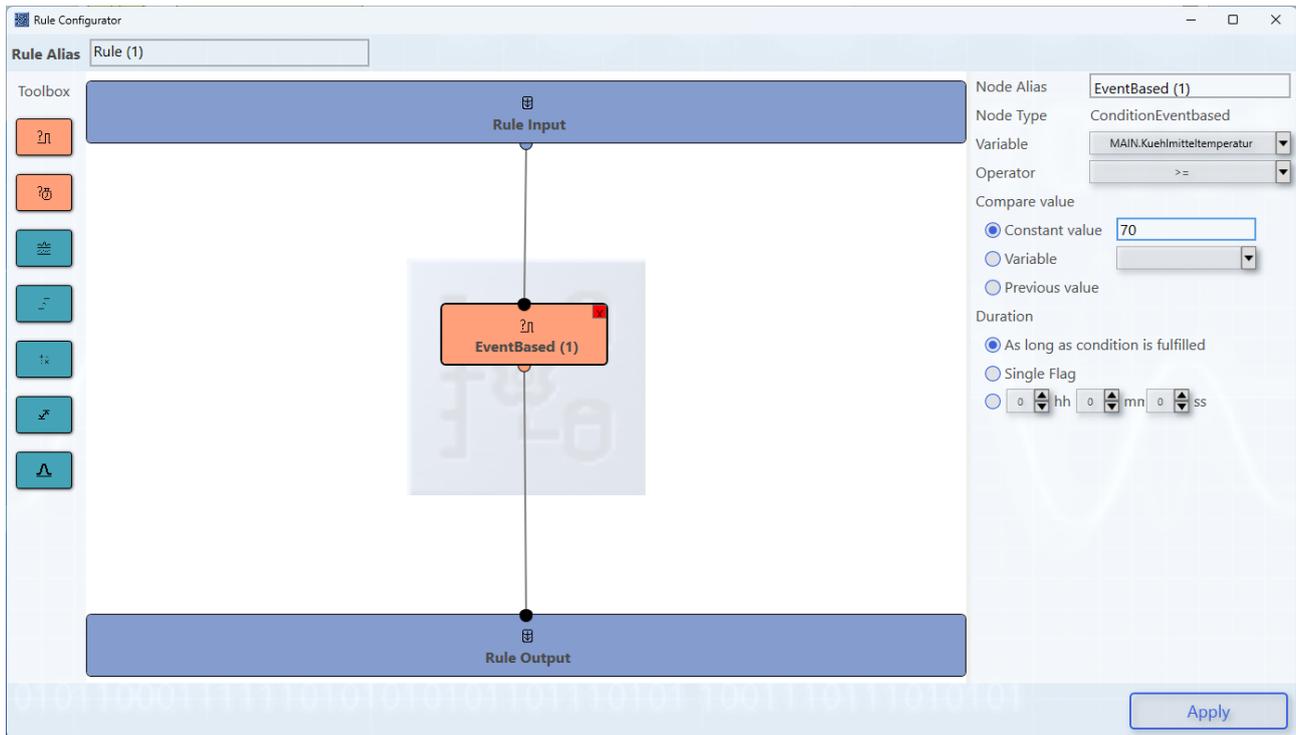
6.2.2.1.4.2.3 Conditions

Condition-Bausteine dienen als Filter und ermöglichen es zeitbasiert oder datenbasiert die eingehenden Daten zu filtern.

6.2.2.1.4.2.3.1 Eventbased

Mit der EventBased-Condition kann innerhalb der Daten nach Daten gefiltert werden, wo ein bestimmtes Ereignis für eingetreten sein muss. Folgende Konfigurationen können vorgenommen werden:

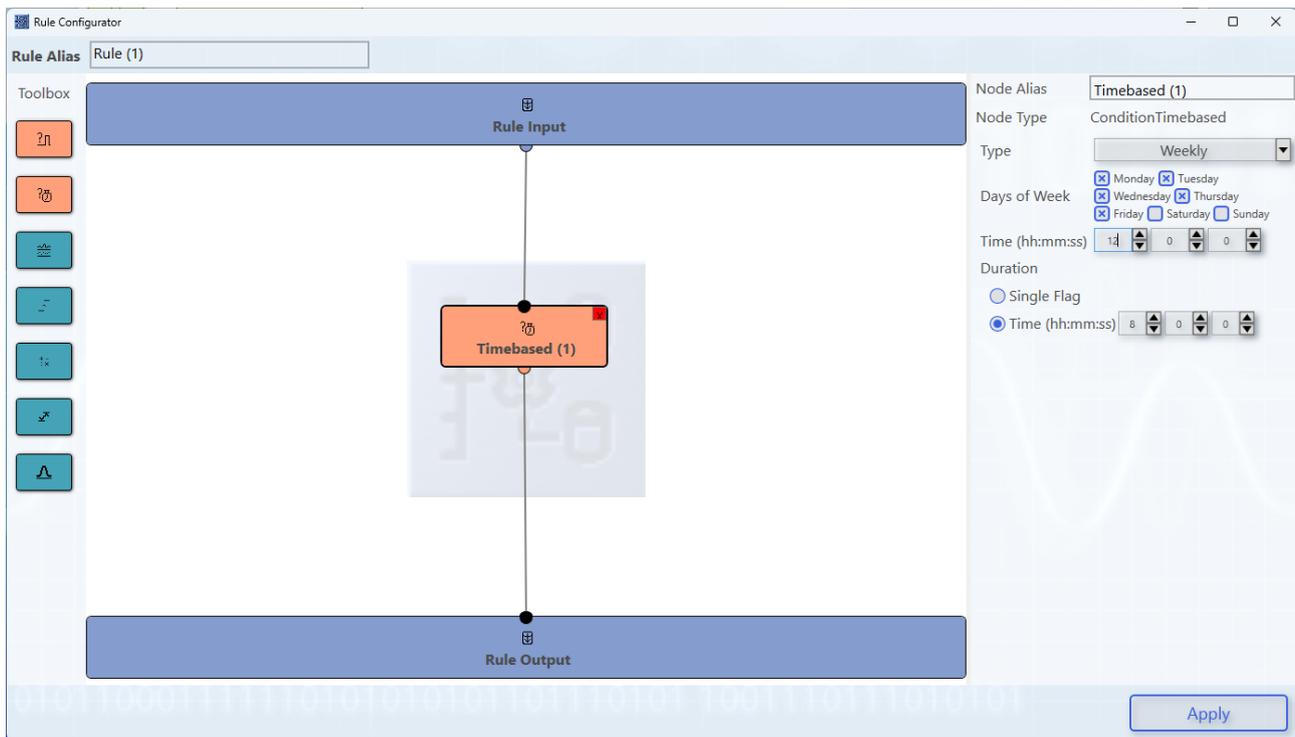
- **Variable:** Auswahl der Variablen, auf welche die Bedingung zutreffen muss.
- **Operator:** Auswahl des Operators für die Bedingung
- **Compare value:** Auswahl des Vergleichswerts
 - **Constant value:** Vergleich mit einem konstanten Wert
 - **Variable:** Vergleich mit einer anderen Variable
 - **Previous value:** Vergleich mit dem vorherigen Wert der ausgewählten Variable.
- **Duration:** Auswahl, wie lange die Daten weitergegeben werden sollen, wenn die Condition eingetreten ist.
 - **As long as condition is fulfilled:** Die Daten werden so lange weitergegeben, wie die Bedingung erfüllt ist.
 - **Single Flag:** Die Daten werden einmalig weitergegeben, wenn die Condition erfüllt ist, also wie bei einer steigenden Flanke.
 - Alternativ können die Daten auch für eine bestimmte Zeit weitergegeben werden, diese kann manuell konfiguriert werden.



6.2.2.1.4.2.3.2 Timebased

Mit der Timebased-Condition kann zeitlich innerhalb der Daten gefiltert werden. Die zugrunde liegende Zeit ist dabei der jeweilige Zeitstempel der Daten. Folgende Konfigurationen können vorgenommen werden:

- **Type:** Auswahl des Typs der Timebased-Condition. Drei Typen stehen zur Auswahl
 - **Interval:** Die Bedingung ist erfüllt, sobald das Intervall erreicht ist.
 - **Weekly:** Die Bedingung ist an bestimmten Wochentagen (Day of Weeks) zu einer bestimmten Uhrzeit (Time) erfüllt.
 - **Monthly:** Die Bedingung ist in bestimmten Monaten (Month) an einem bestimmten Tag (Day) zu einer bestimmten Uhrzeit (Time) erfüllt.
- **Duration:** Auswahl, wie lange die Daten weitergegeben werden sollen, wenn die Condition eingetreten ist
 - **Single Flag:** Die Daten werden einmalig weitergegeben, wenn die Condition erfüllt ist, also wie bei einer steigenden Flanke.
 - **Time:** Alternativ können die Daten auch für eine bestimmte Zeit weitergegeben werden, diese kann manuell konfiguriert werden.



6.2.2.1.4.2.4 Actions

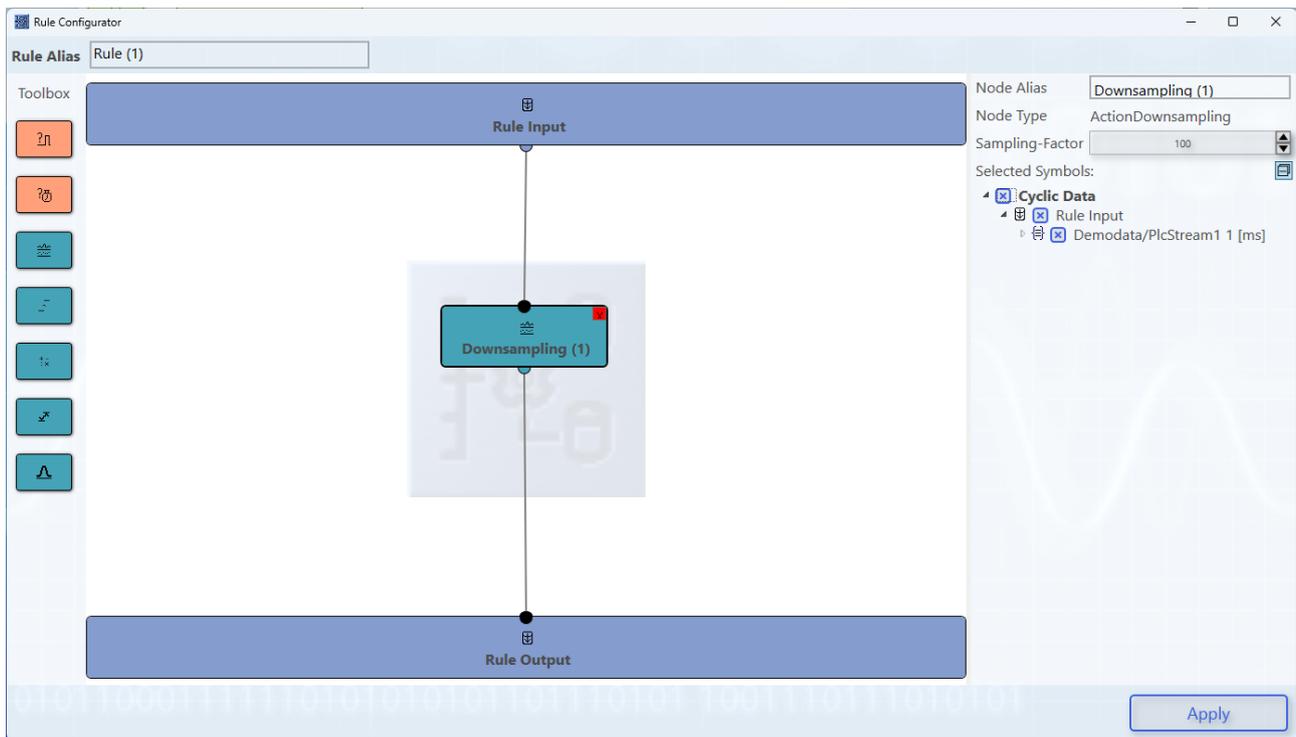
Action-Bausteine bieten die Möglichkeit zur Aggregation der Daten (wie bspw. die Berechnung von Min/Max/ Avg oder weiteren Operationen).

6.2.2.1.4.2.4.1 Downsampling

Konfiguration eines Downsamplings einzelner Variablen. Der Sampling-Faktor gibt die Sampling Rate an. Im Bereich Selected-Symbols kann man die Variablen auswählen, auf die das Downsampling angewendet werden soll. Möchten Sie alle Daten einer Rule sampeln, können Sie dies im entsprechenden Recording konfigurieren.

Für jedes ausgewählte Symbol werden folgende Symbole bereitgestellt:

- `_Sampled`: Gesampeltes Symbol

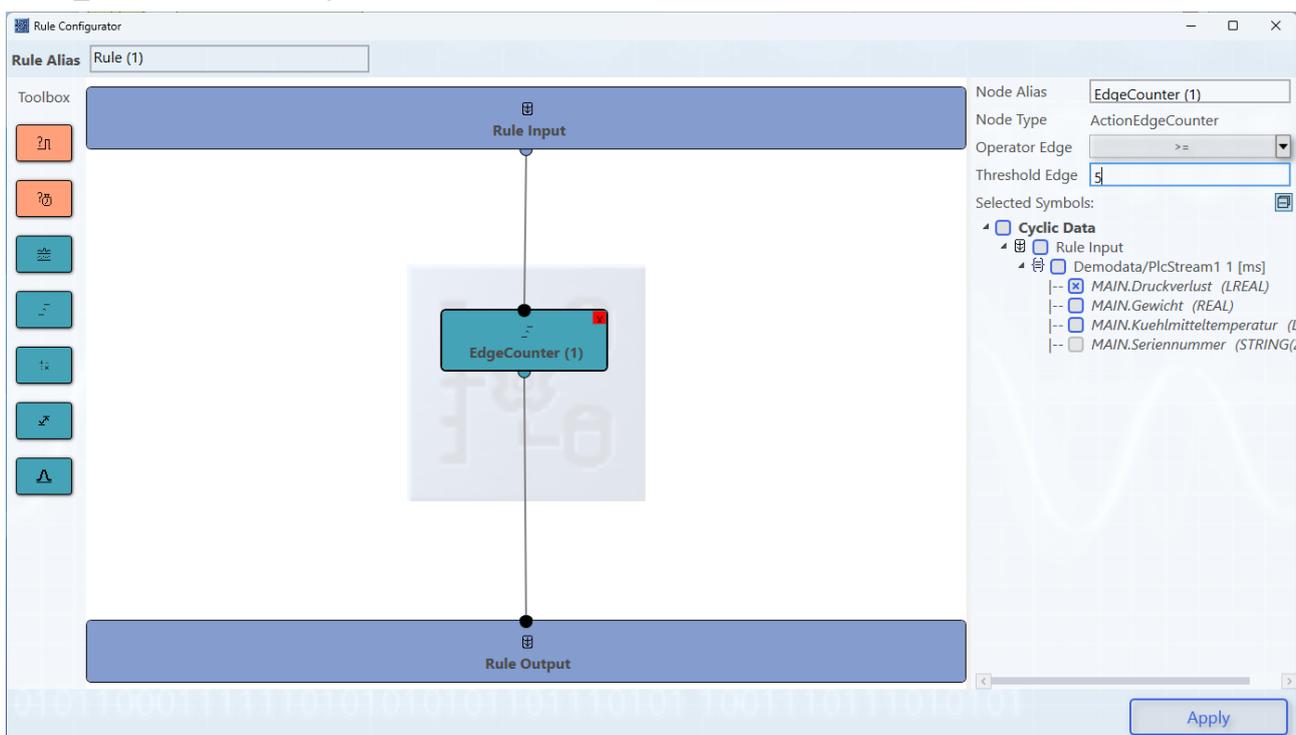


6.2.2.1.4.2.4.2 EdgeCounter

Bestimmt die Flanken innerhalb der Daten. Zu konfigurieren ist hierfür der Operator Edge und der zu vergleichende Schwellwert. Im Bereich Selected-Symbols kann man die Variablen auswählen, auf welche der Edge-Counter angewendet werden soll.

Für jedes ausgewählte Symbol werden folgende Symbole bereitgestellt:

- `_Count`: Anzahl der gezählten Flanken
- `_Edge`: Boolean ob eine Flanke erkannt ist
- `_LastEvent`: Zeitstempel des letzten Events

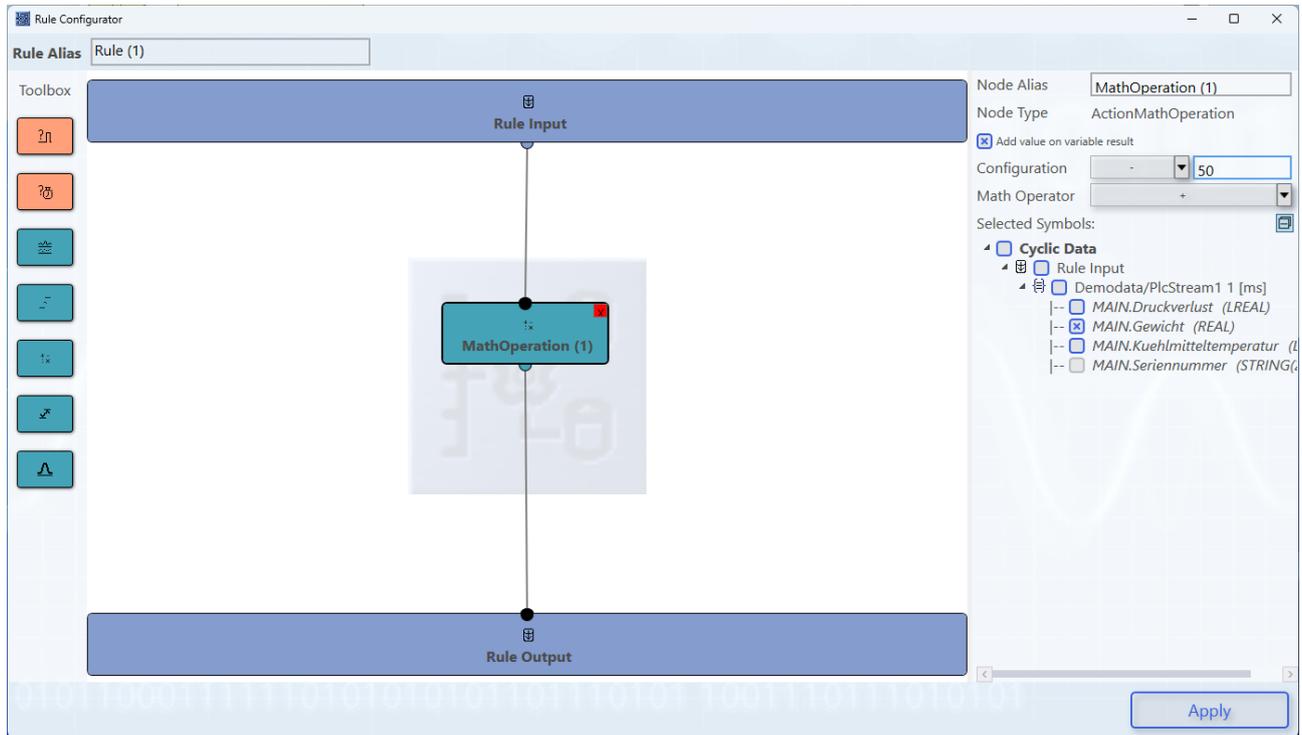


6.2.2.1.4.2.4.3 MathOperation

Mit dem MathOperation-Baustein können Symbole miteinander verrechnet werden. Die miteinander zu verrechnenden Symbole können unter **Selected Symbols** ausgewählt werden. Der Operator lässt sich unter **MathOperator** auswählen. Soll das Ergebnis mit einem zusätzlichen konstanten Wert verrechnet werden, aktivieren Sie die Checkbox **Add value on variable result**. Im Bereich **Configuration** können Sie den Operator und den konstanten Wert, mit welchem das Ergebnis der Symbole verrechnet werden soll, konfigurieren.

Es wird folgendes Symbol bereitgestellt:

- **Result:** Ergebnis der Berechnung

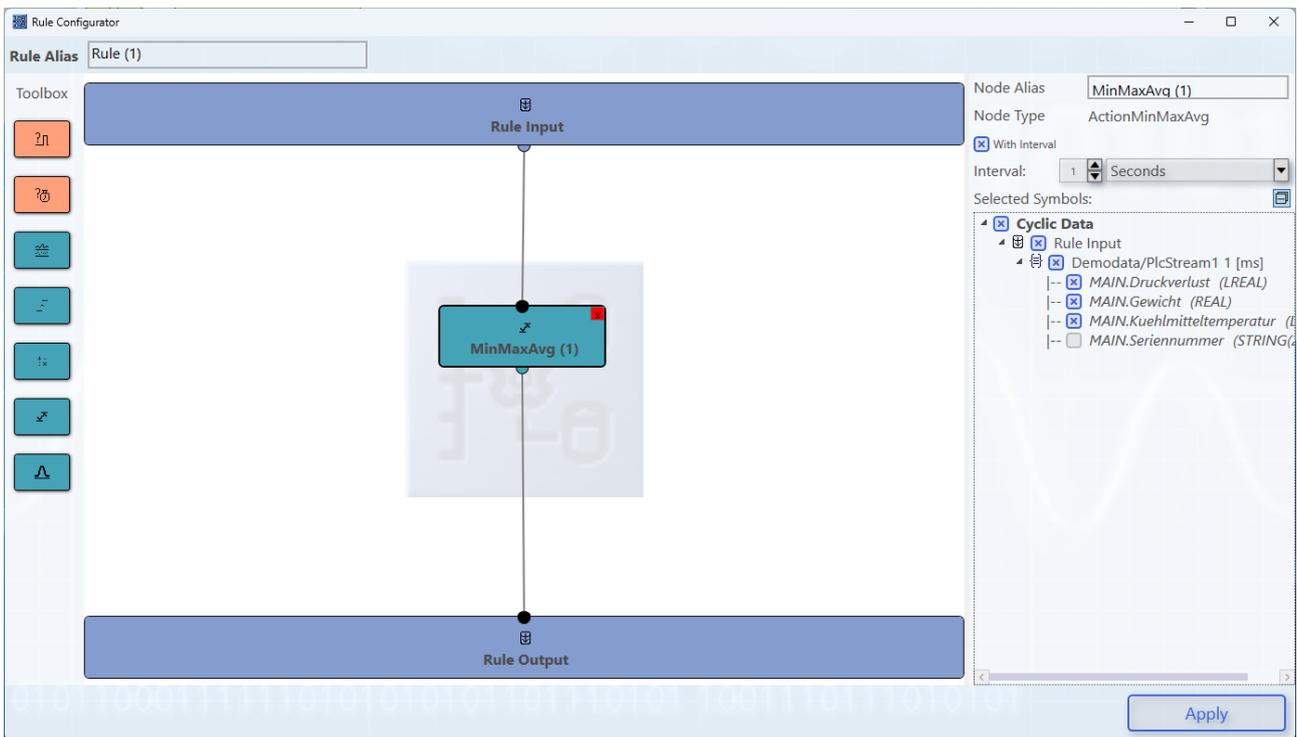


6.2.2.1.4.2.4.4 Min/Max/Avg

Berechnung des Minimal-/Maximal- und Durchschnittswerts für ausgewählte Symbole. Im Bereich **Selected Symbols** können Sie die Variablen auswählen, für welche die Werte berechnet werden sollen. Aktivieren Sie die Checkbox **With Interval** kann ein Intervall, für welches die Werte berechnet werden sollen, konfiguriert werden.

Für jedes ausgewählte Symbol werden folgende Symbole bereitgestellt:

- **_Min:** Minimum des Symbols
- **_Max:** Maximum des Symbols
- **_Avg:** Durchschnitt des Symbols

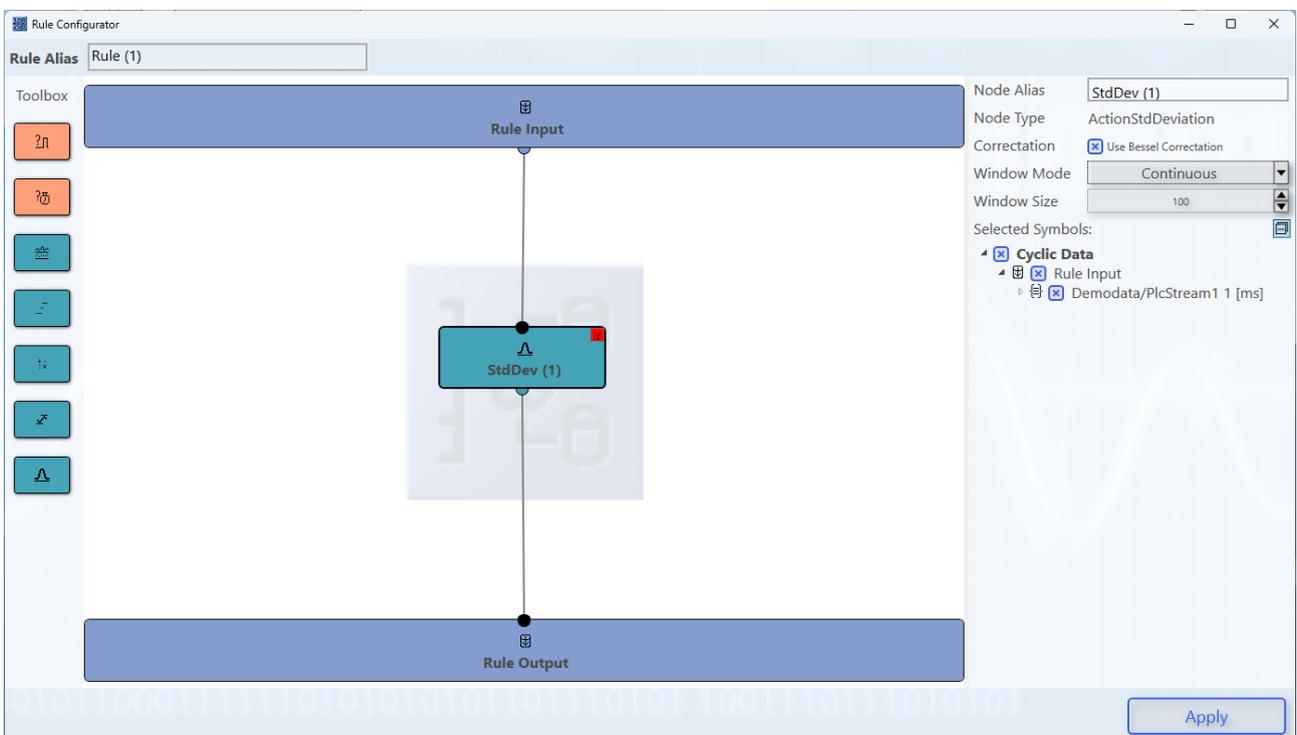


6.2.2.1.4.2.4.5 StandardDeviation

Berechnung der Standardabweichung für ausgewählte Symbole. Im Bereich **Selected Symbols** können Sie die Variablen auswählen, für welche die Werte berechnet werden sollen. Die Berechnung der Standardabweichung kann entsprechend des Bausteins konfiguriert werden.

Für jedes ausgewählte Symbol werden folgende Symbole bereitgestellt:

- **_STD**: Standardabweichung des Symbols



6.2.2.1.4.2.5 Trigger

Für eine BatchDataRule (basiert auf Daten aus einem Recording) können Trigger konfiguriert werden. Diese Trigger ermöglichen es, dass die Rule zeitbasiert oder eventgesteuert aufstartet. Zudem kann eine Rule jederzeit manuell getriggert werden. Die Konfiguration der Trigger erfolgt im Rule Trigger-Konfigurator. Dieser kann über das Property-Fenster der Rule geöffnet werden. Im Rule Trigger-Konfigurator können die zeitbasierten und eventbasierten Trigger konfiguriert werden. Derzeit können lediglich zeitbasierte oder eventbasierte Trigger für eine Rule verwendet werden.

The screenshot displays the RuleEngine Pipeline Editor interface. At the top, the Pipeline Alias is 'MyRuleEnginePipelineAlias' and the Storage Provider is 'TwinCAT Analytics StorageProvider ("INGMARR-NB02")'. The main workspace shows a pipeline diagram with the following components and connections:

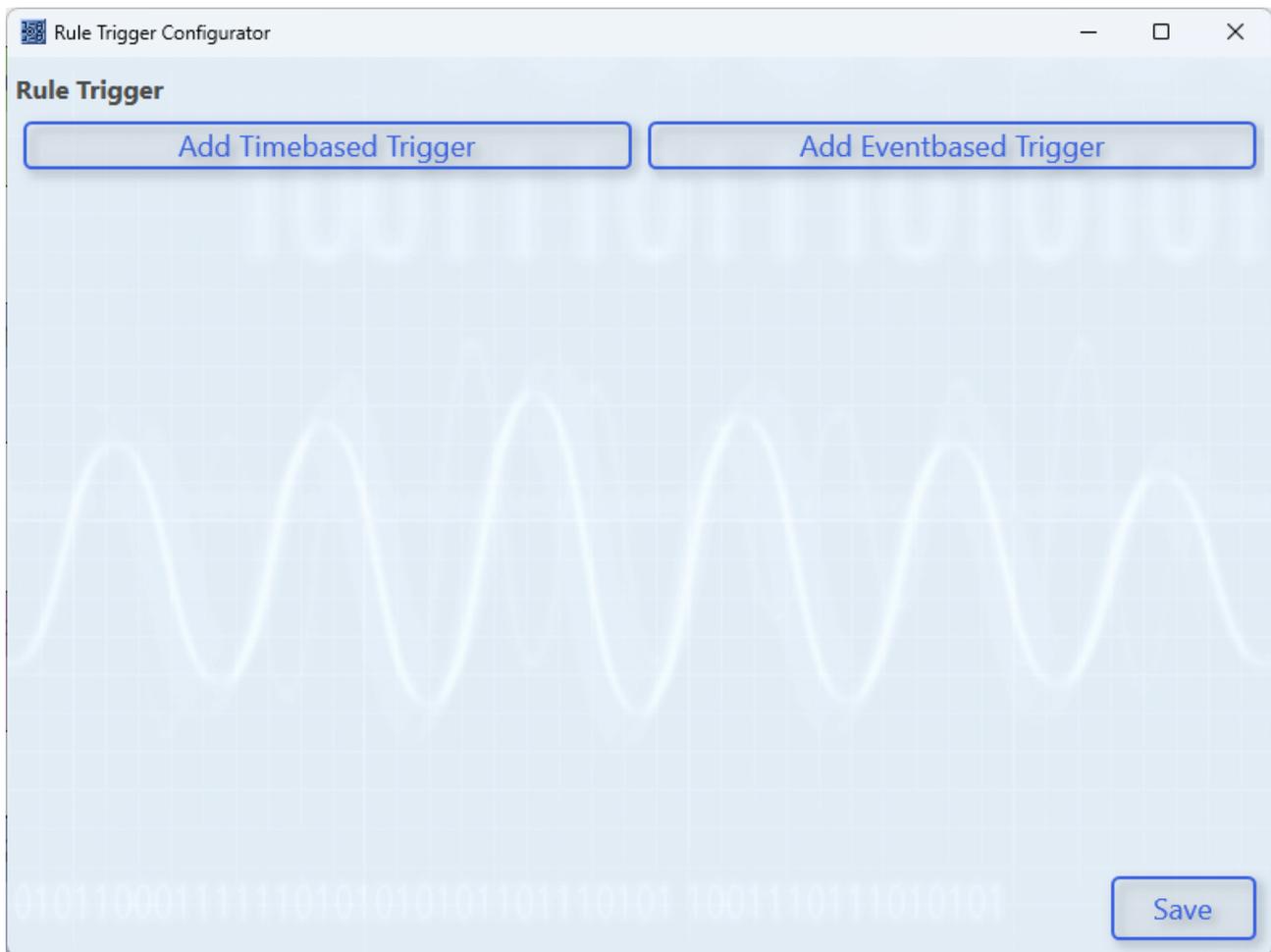
- MQTT Source (1)** (green box) connects to **Rule (1)** (blue box).
- Rule (1)** connects to both **Recording (1)** (pink box) and **Recording (2)** (pink box).
- Recording (1)** and **Recording (2)** both connect to **Rule (2)** (blue box) and **Rule (3)** (blue box).

Each component has a 'NotInitialized' status. A 'Save' button is located at the bottom right of the editor. On the right side, the **Rule (3) - Properties** window is open, showing the following details:

Rule (3) - Properties	
Alias	Rule (3)
Type	BatchDataRule
Nodes	2
Symbols	0

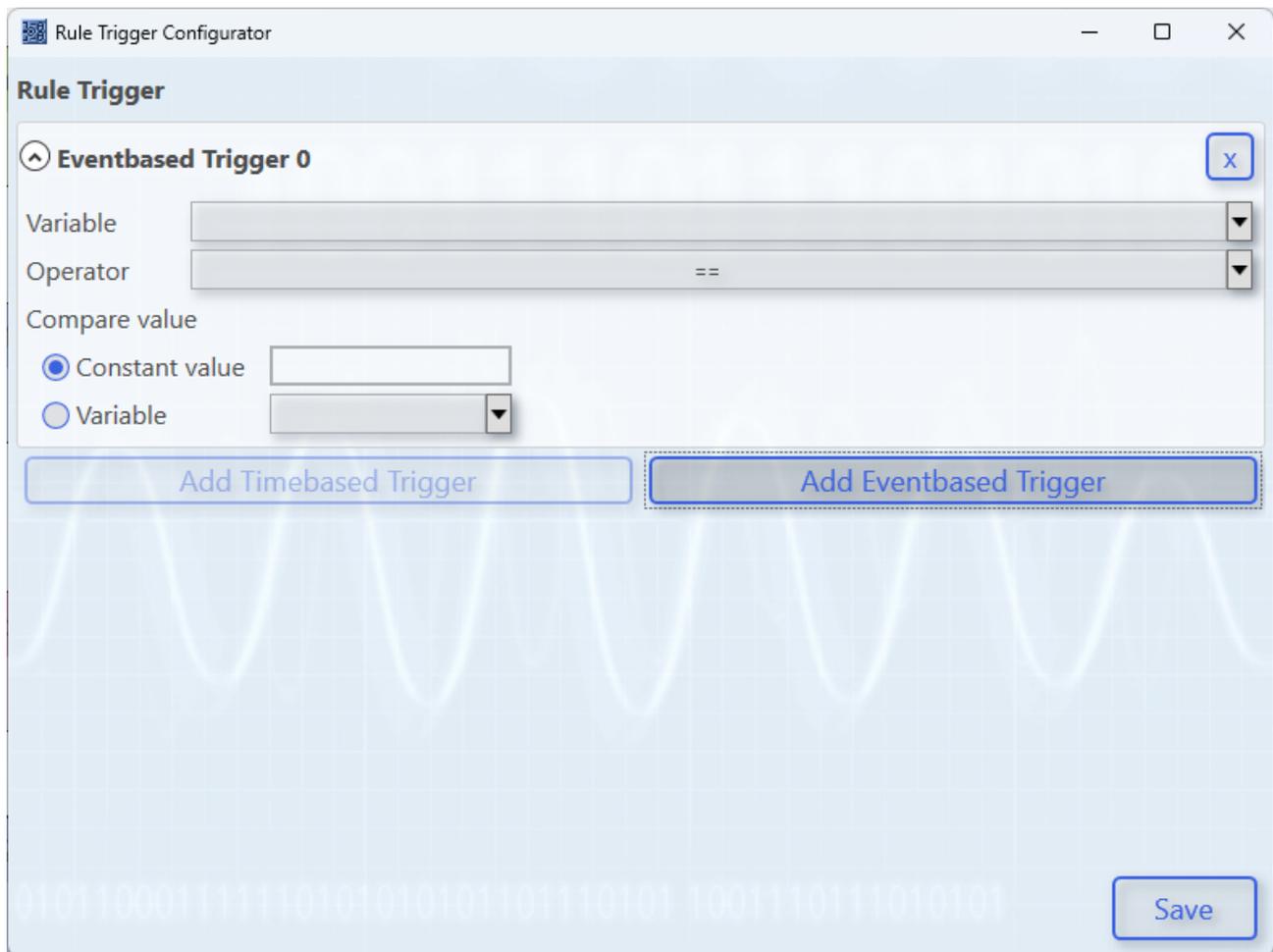
Below the properties, the **Rule Trigger** section is visible, showing:

Rule Trigger	
Trigger	0



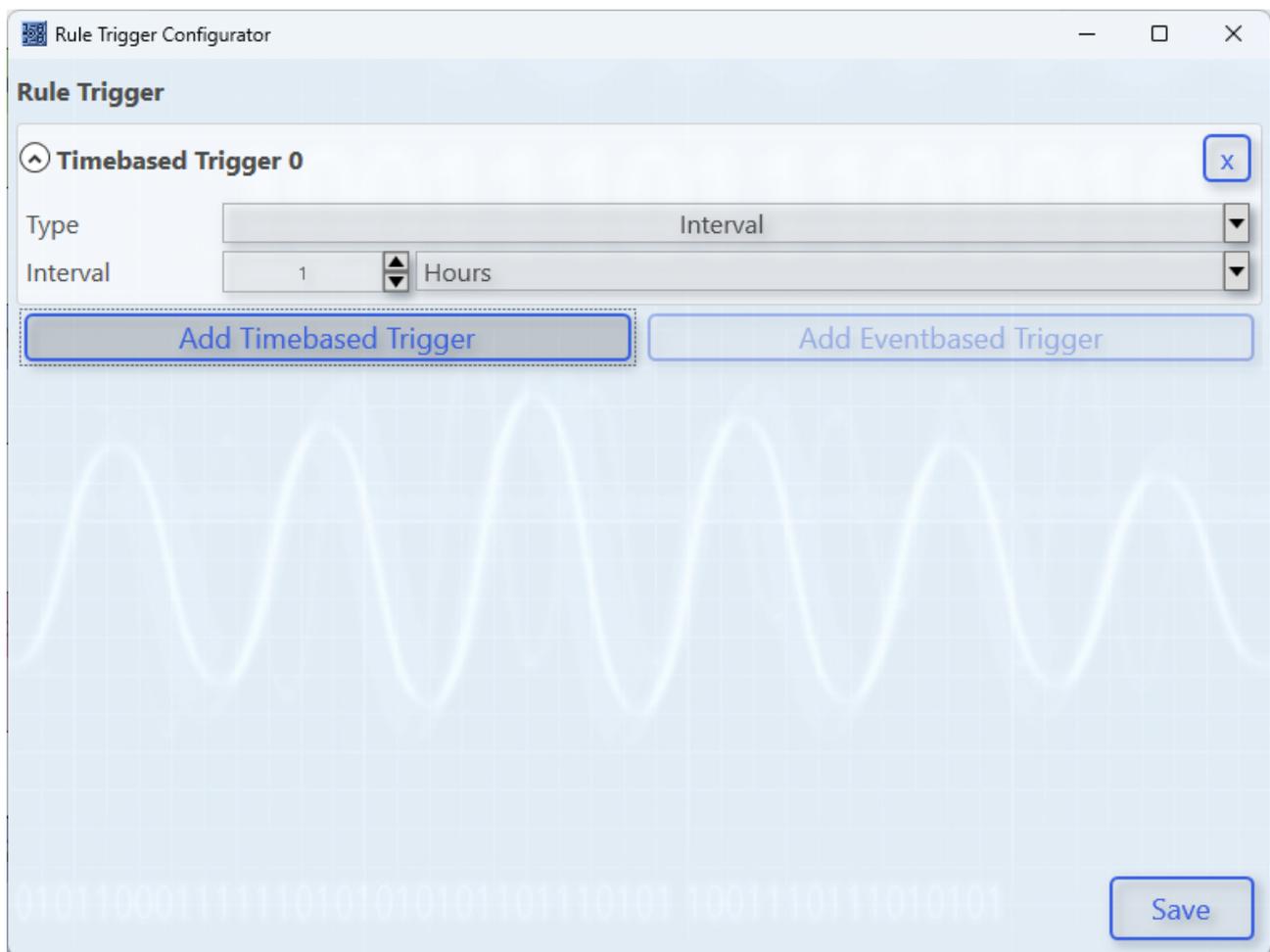
6.2.2.1.4.2.5.1 Eventbased

Die Konfiguration eines eventbasierten Triggers entspricht einer [Eventbased-Condition \[► 79\]](#) innerhalb der Rule. Die zugrundeliegenden Daten sind hierbei die Daten, welche gerade in das Recording, welche als Datenquelle für die Rule gilt, geschrieben werden. Eine Konfiguration der Duration ist nicht möglich und notwendig. Wird die Trigger Bedingung erfüllt, läuft die Rule so lange, bis die Daten entsprechend prozessiert sind und deaktiviert sich anschließend bis zum nächsten Trigger.



6.2.2.1.4.2.5.2 Timebased

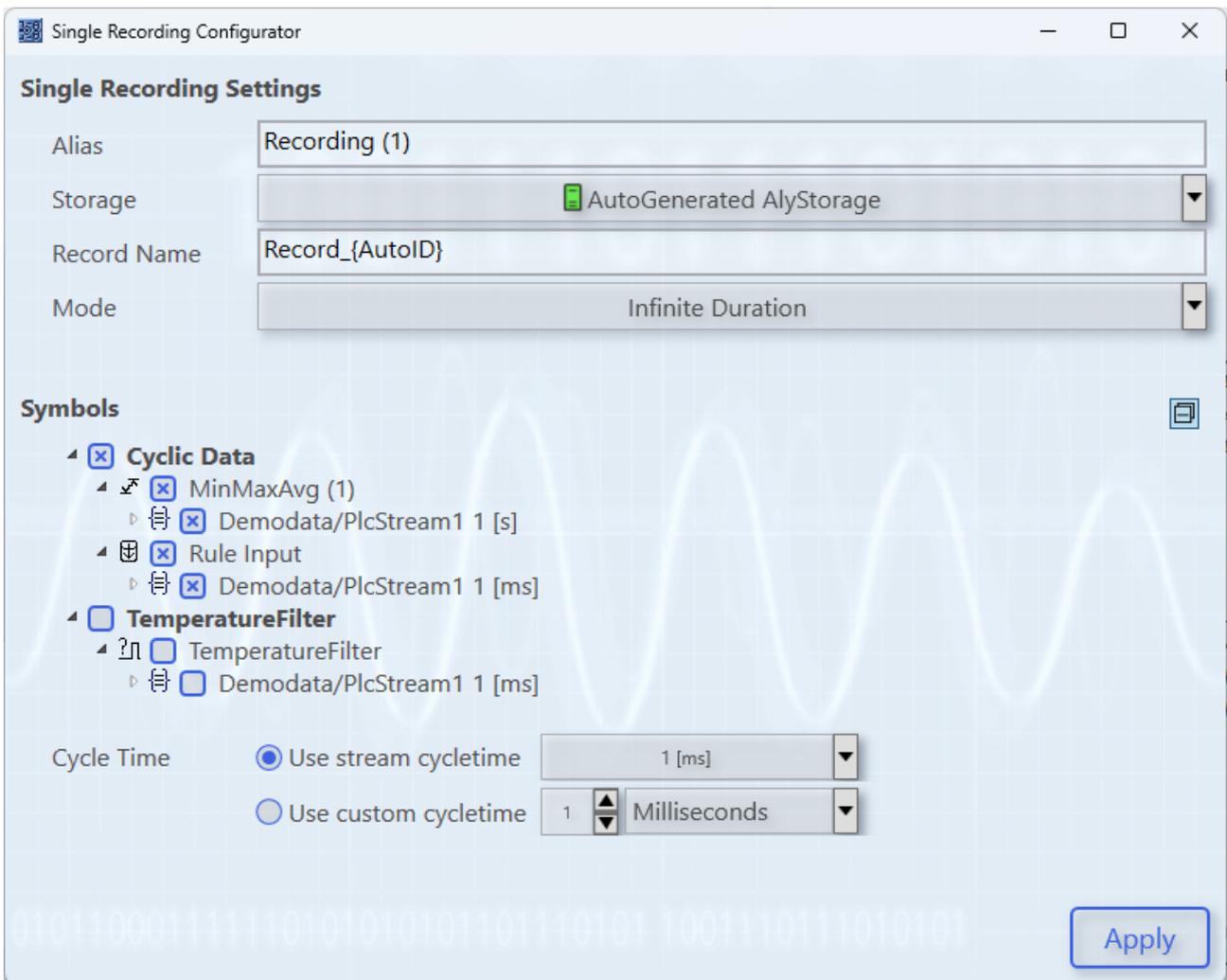
Die Konfiguration eines zeitbasierten Triggers entspricht einer [Timebased-Condition](#) [► 80] innerhalb der Rule. Die zugrundeliegende Zeit ist hierbei die Systemzeit des Geräts, auf welchem der Storage Provider läuft. Eine Konfiguration der Duration ist nicht möglich und notwendig. Wird die Trigger Bedingung erfüllt, läuft die Rule so lange, bis die Daten entsprechend prozessiert sind und deaktiviert sich anschließend bis zum nächsten Trigger.



6.2.2.1.4.3 Bearbeiten von Recordings

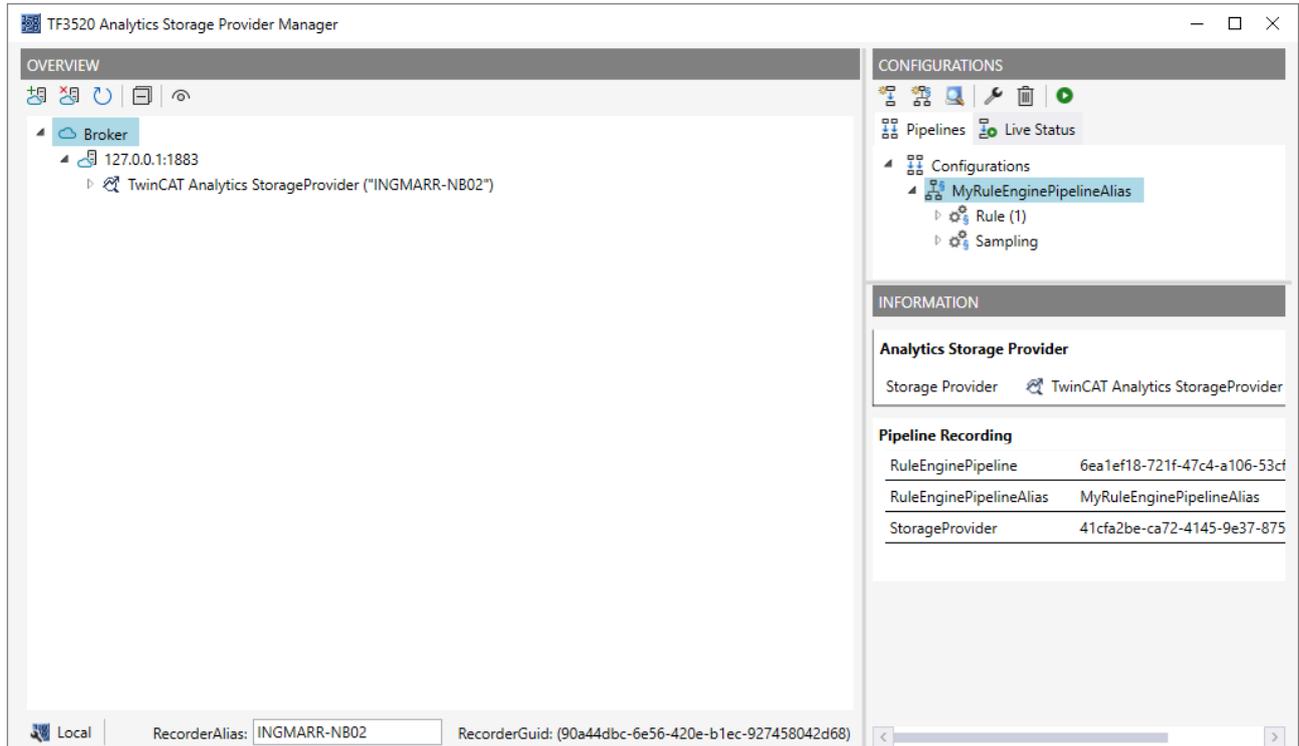
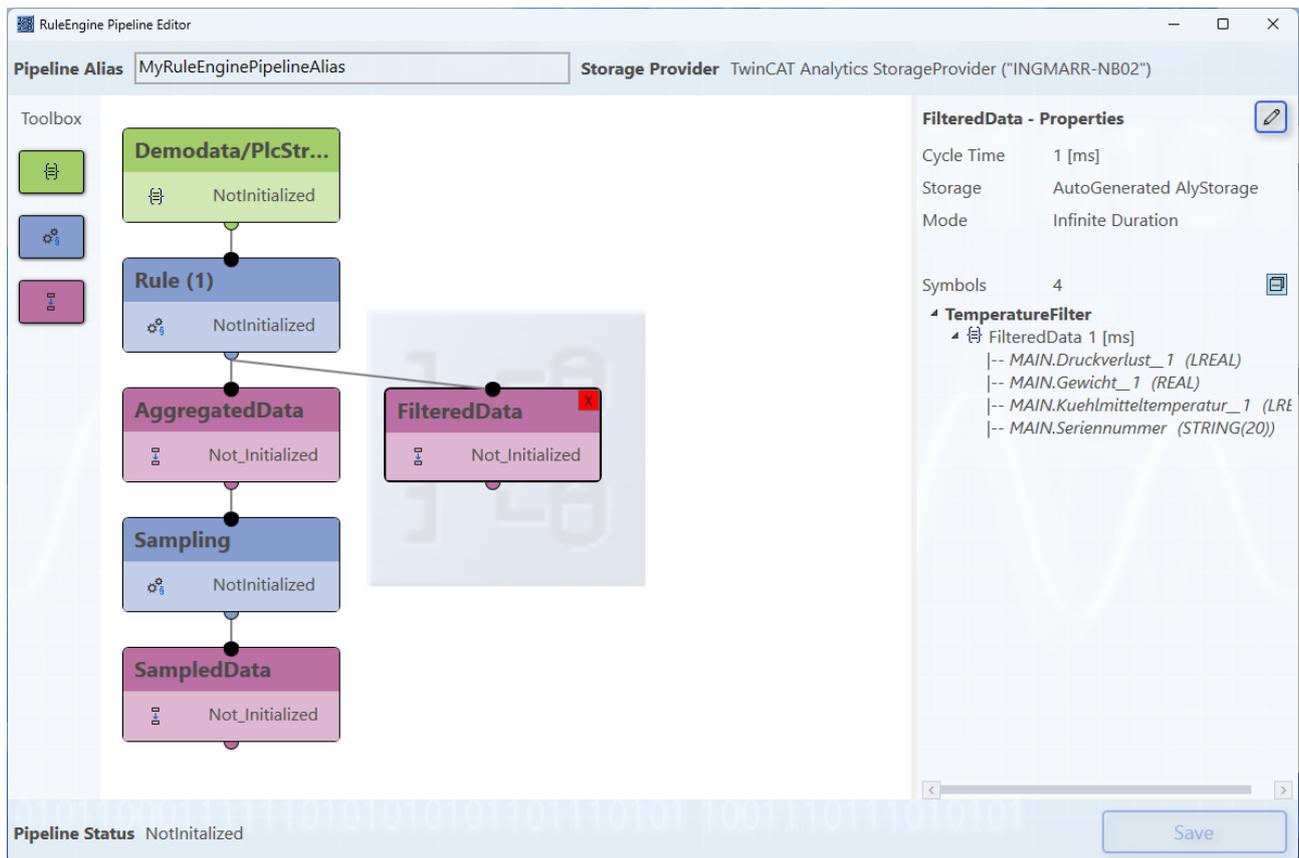
Die Bearbeitung von Recordings erfolgt im Single Recording Configurator. Hier stehen folgende Einstellungen zur Verfügung:

- **Alias:** Recording Alias
- **Storage:** Datensenke für das Recording (wird im Storage Provider Configurator konfiguriert)
- **Record Name:** Name des Records
- **Mode:** Modus des Records
 - **Infinite Duration:** Der Record läuft so lange, wie die RuleEngine-Pipeline läuft und Daten für den Record vorliegen.
 - **Timebased Ringbuffer:** Der Record läuft so lange, wie die RuleEngine-Pipeline läuft und Daten für den Record vorliegen. Die Daten werden in einem zeitlichen begrenzten Ringspeicher gehalten.
 - **Databased Ringbuffer:** Der Record läuft so lange, wie die RuleEngine-Pipeline läuft und Daten für den Record vorliegen. Die Daten werden in einem speichergrößentechnisch begrenzten Ringspeicher gehalten.
- **Symbols:** Auswahl der Symbole, welche mit dem Recording abgespeichert werden sollen. Sie haben dabei die Auswahl zwischen den zyklischen Daten oder anhand von Conditions gefilterten Daten.
- **CycleTime:** Auswahl der Zykluszeit für das Recording. Diese Auswahl kann nur vorgenommen werden, wenn Sie zyklische (ungefilterte) Daten abspeichern wollen. Sie haben dabei die Möglichkeit, eine Zykluszeit basierend auf den Variablen oder eine eigene Zykluszeit für das Recording zu konfigurieren. Beachten Sie, dass Daten verloren gehen, wenn die Zykluszeit größer als die minimale Zykluszeit der Daten ist. In diesem Fall erscheint eine Meldung.



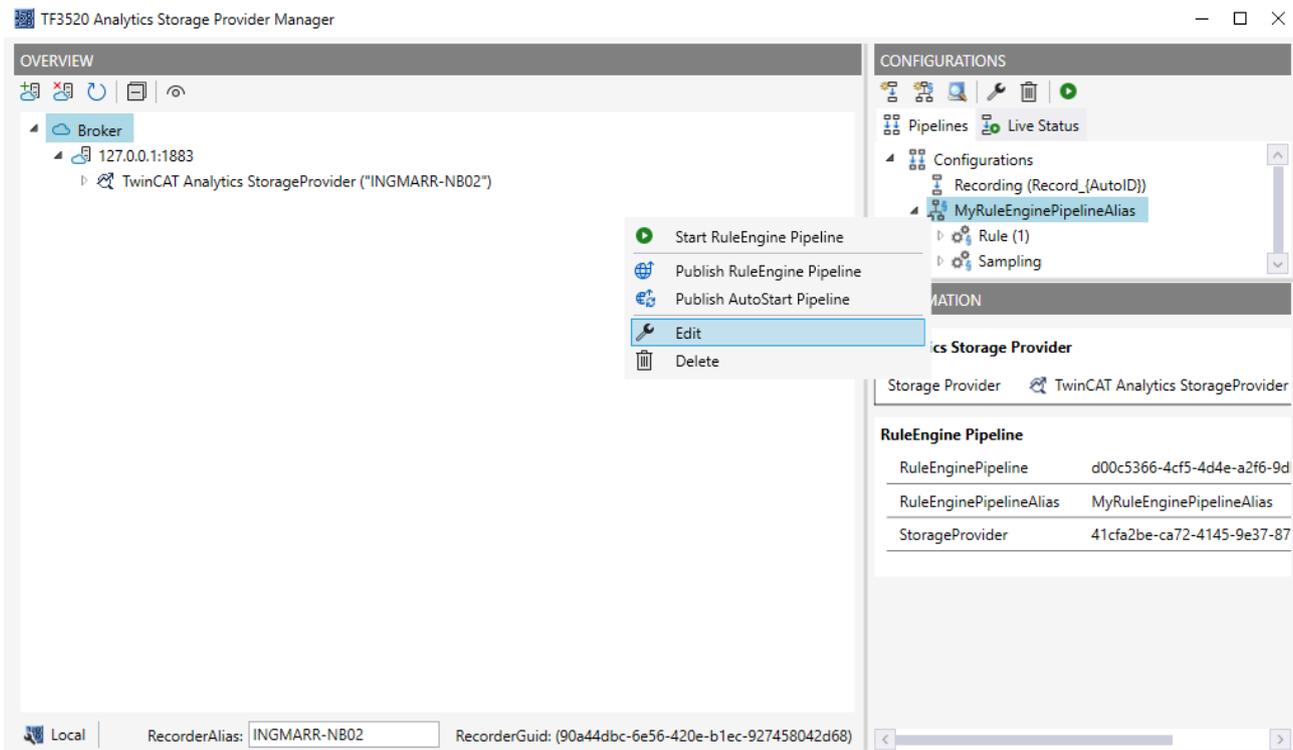
6.2.2.1.5 Speichern einer RuleEngine-Pipeline

Eine konfigurierte RuleEngine-Pipeline kann über den Button **Save** (unten rechts) im RuleEngine-Pipeline-Editor gespeichert werden. Das Speichern ist nur möglich, wenn die Konfiguration fehlerfrei ist. Eventuelle Fehler werden über eine Meldung ausgegeben und die fehlerhaft konfigurierten Bausteine werden grafisch hervorgehoben.



6.2.2.1.6 Nachträgliches Bearbeiten einer RuleEngine-Pipeline

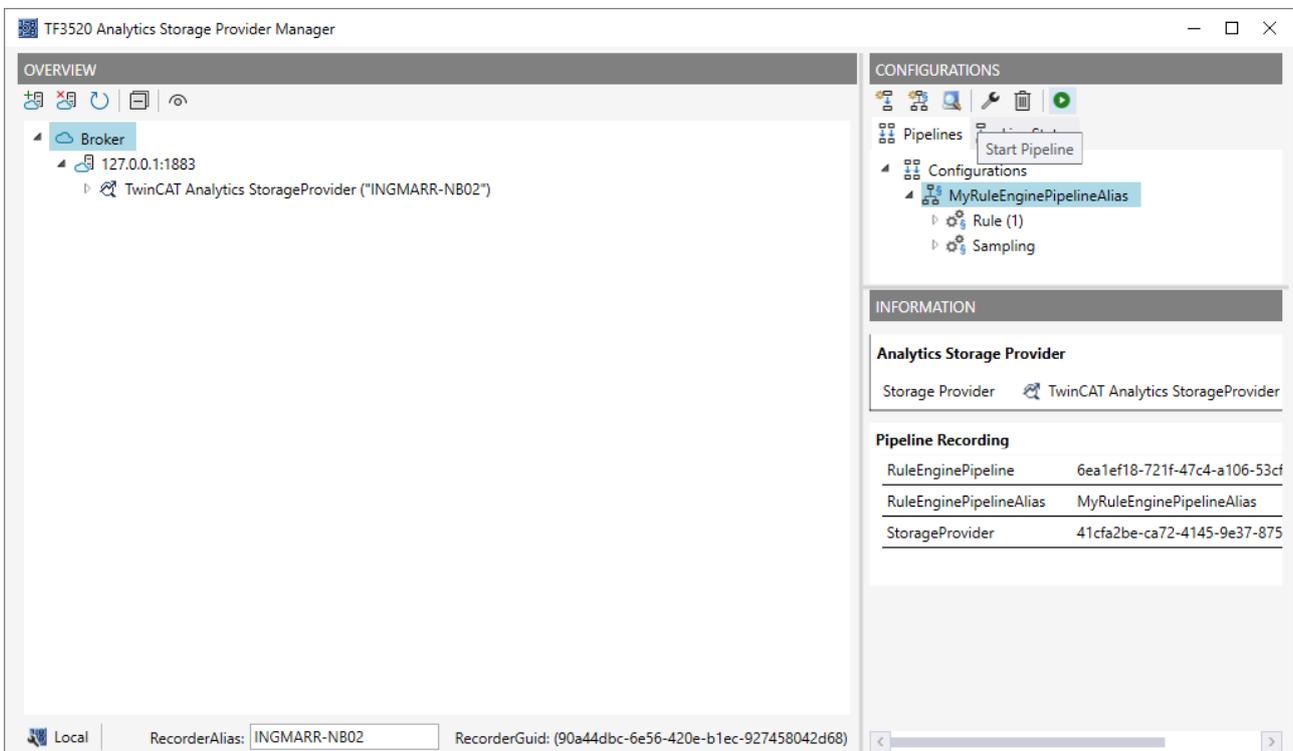
Eine angelegte RuleEngine-Pipeline kann über einen Rechtsklick im Kontextmenü **Edit** bearbeitet und modifiziert werden. Beachten Sie, dass vorhandene Daten nicht länger in der Pipeline verwendet werden können, wenn die in der Pipeline enthaltenen Recordings angepasst werden.



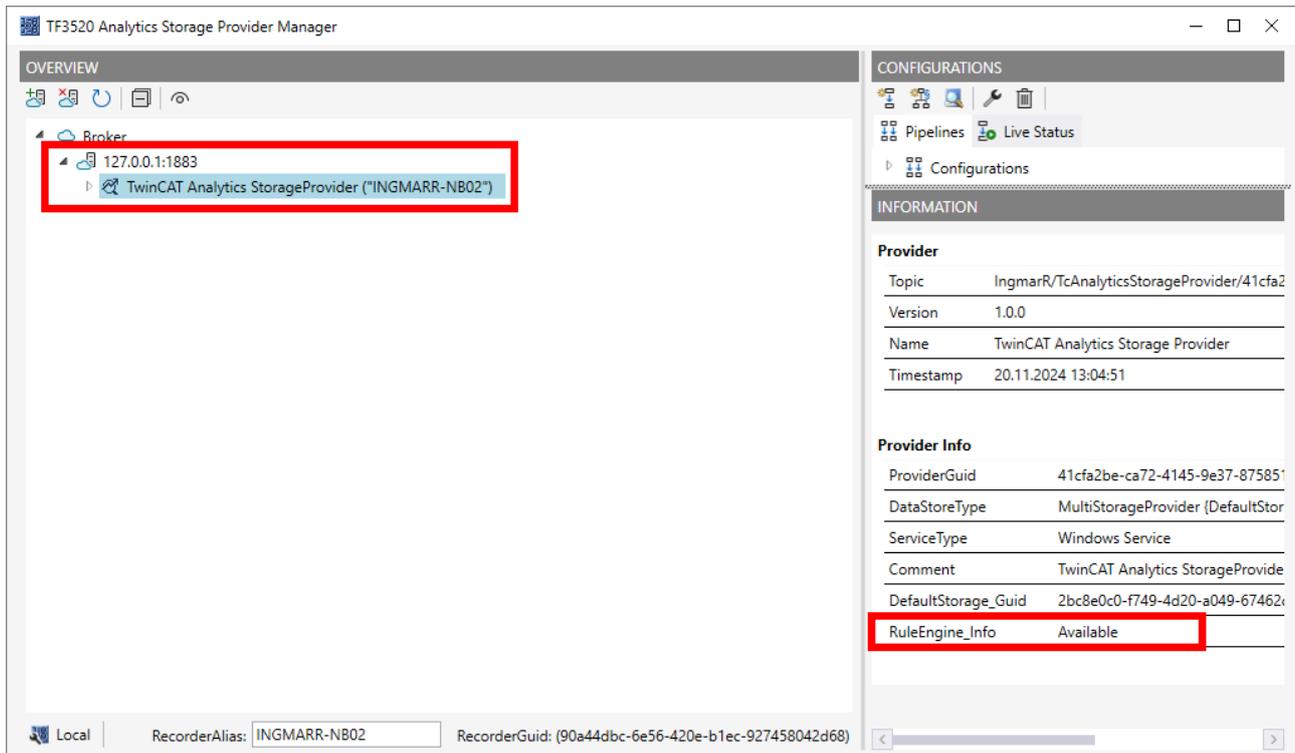
6.2.2.2 Anwendung

6.2.2.2.1 Starten/Stoppen einer RuleEngine-Pipeline

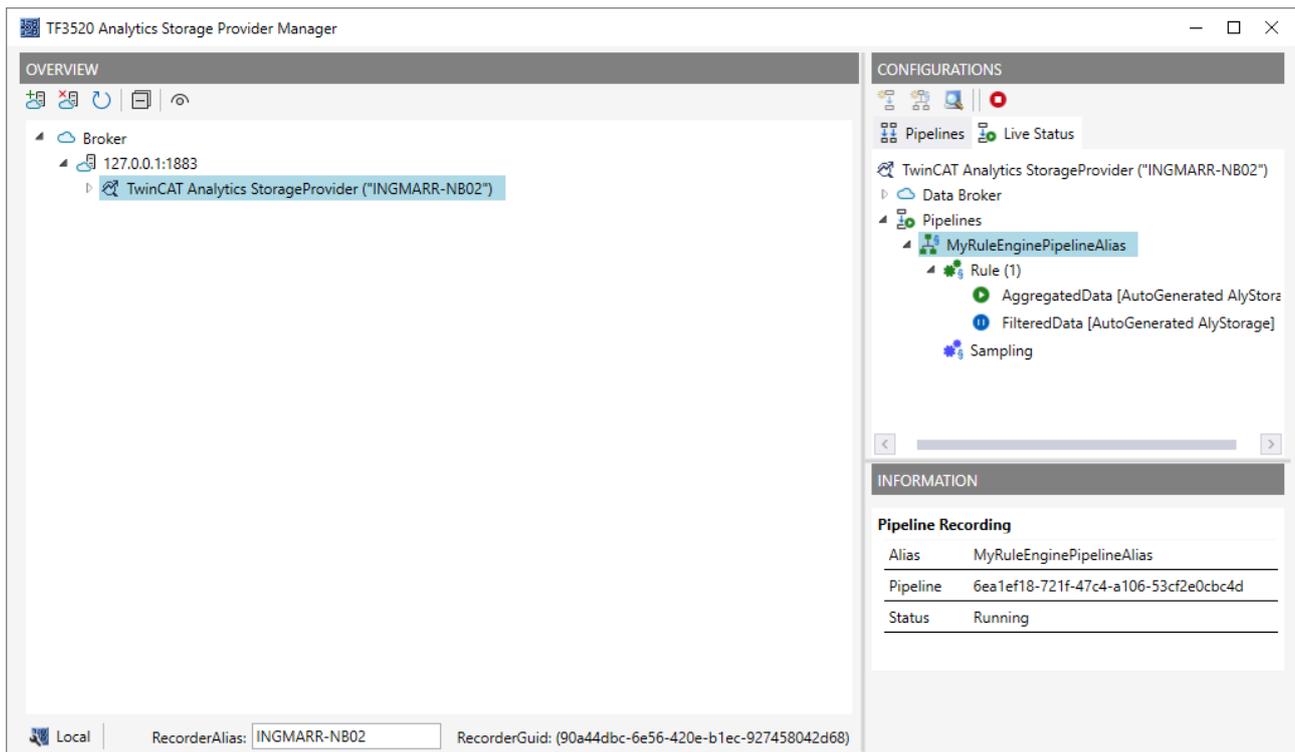
Eine konfigurierte RuleEngine-Pipeline kann über das Icon **Start** gestartet werden.



Ein Start ist nur möglich, wenn der Storage Provider läuft und die im Storage Provider befindliche RuleEngine fehlerfrei läuft. Der Status der RuleEngine kann über die Informationen des jeweiligen Storage Providers ausgelesen werden.

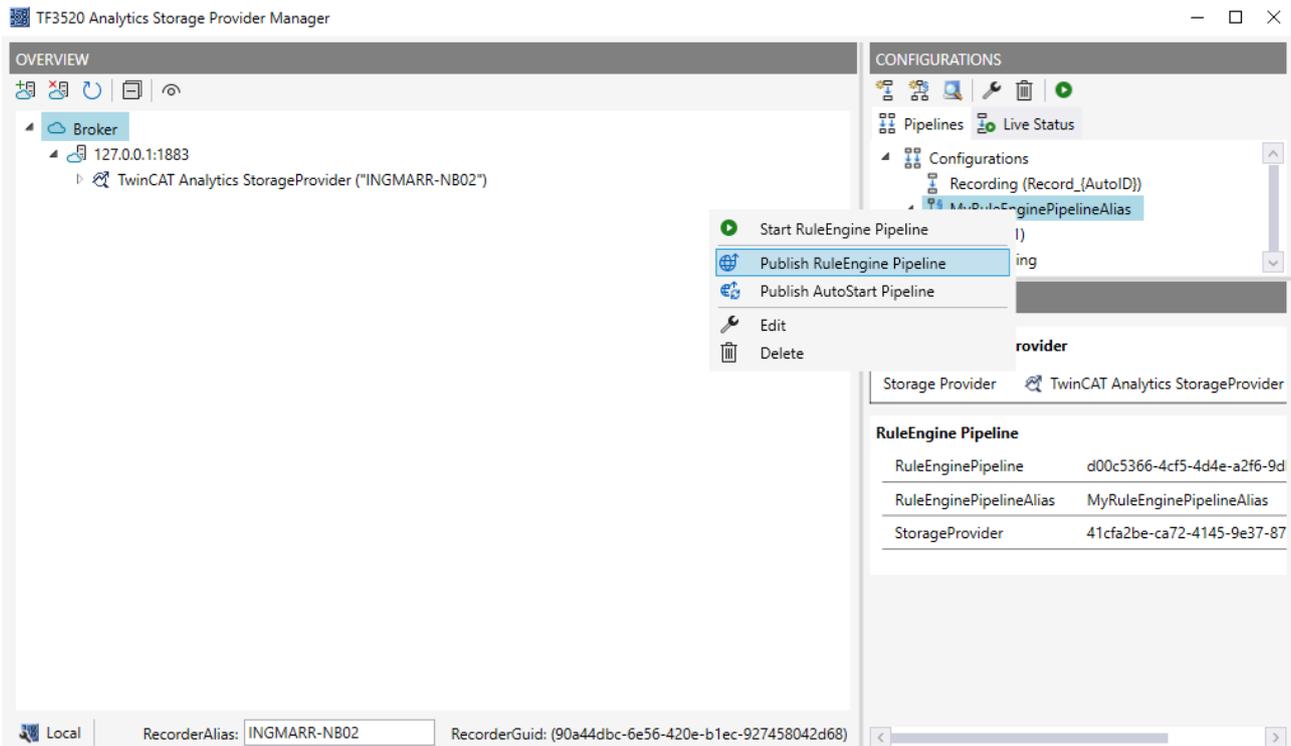


Gestartete RuleEngine-Pipelines können Sie im Live-Status-Tab einsehen und dort auch wieder stoppen.

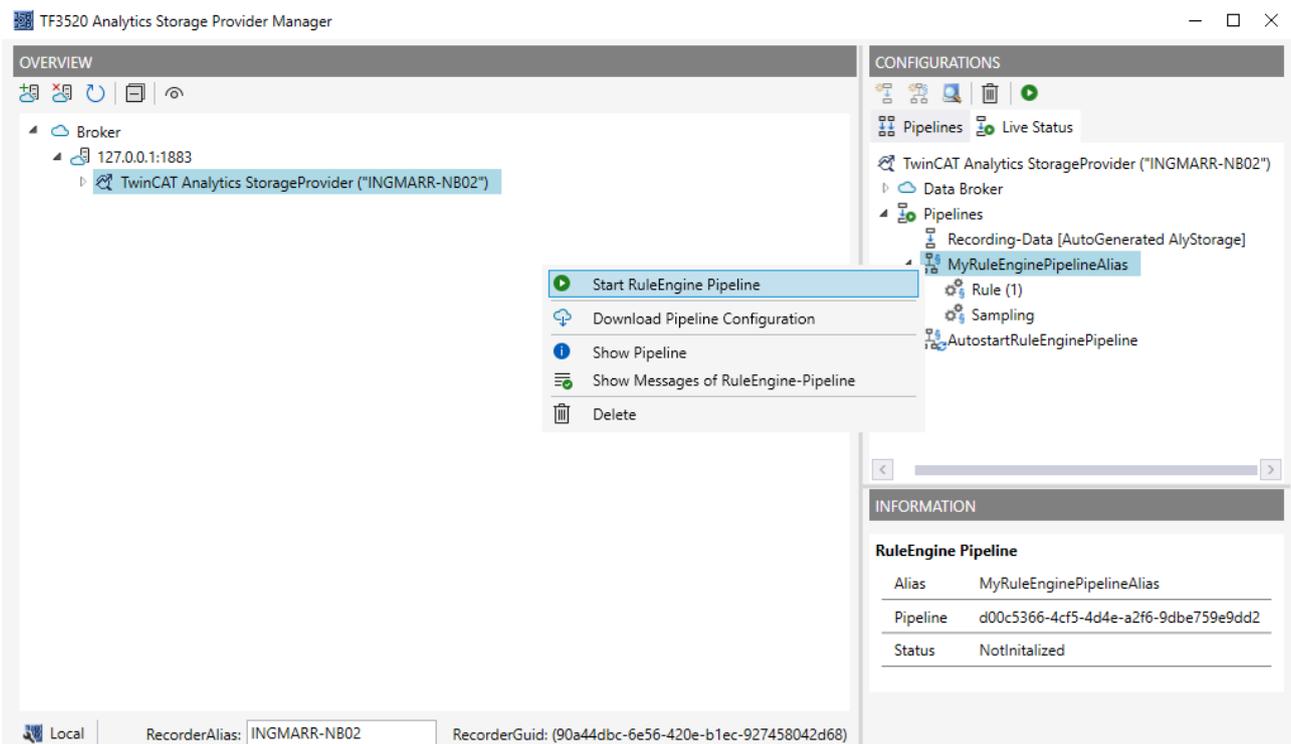


6.2.2.2 Publizieren von RuleEngine-Pipelines

Lokal konfigurierte RuleEngine-Pipelines können an den hinterlegten Storage Provider gepublished werden. Dadurch wird die RuleEngine-Pipeline zusätzlich am Storage Provider gespeichert und ist global verfügbar. Andere Benutzer können diese herunterladen, bearbeiten oder starten. Der Publish erfolgt über Rechtsklick im Kontextmenü **Publish RuleEngine Pipeline** auf die lokal konfigurierte Pipeline. Zusätzlich können RuleEngine-Pipelines im Autostart-Modus gepublished werden. Das bedeutet, dass die Pipelines automatisch nach dem Start des Storage Provider Services gestartet werden.



Unter **Live-Status** sind alle gepublizierten Pipelines einsehbar. Diese können gestartet, gelöscht oder heruntergeladen werden.



6.2.2.2.3 RuleEngine-Pipeline Zustände

Eine RuleEngine-Pipeline kann folgende Zustände haben:

	NotInitialized: Die gestartete RuleEngine-Pipeline wurde am Storage Provider empfangen, jedoch noch nicht verarbeitet
	Initializing: Die RuleEngine-Pipeline wird initialisiert. Die dazugehörigen Rules werden angelegt.
	IsStarting: Die RuleEngine-Pipeline wird aufgestartet. Enthaltene Rules werden aufgestartet.
	Running: Die RuleEngine-Pipeline läuft. Alle Rules befinden sich mindestens im Pending Zustand und es liegt kein Fehler vor.
	IsStopping: Die RuleEngine-Pipeline wird gestoppt, alle Rules fahren herunter.
	Stopped: Die RuleEngine-Pipeline ist gestoppt, alle Rules sind heruntergefahren.
	Error: Bei mindestens einer Rule liegt ein Fehler vor.

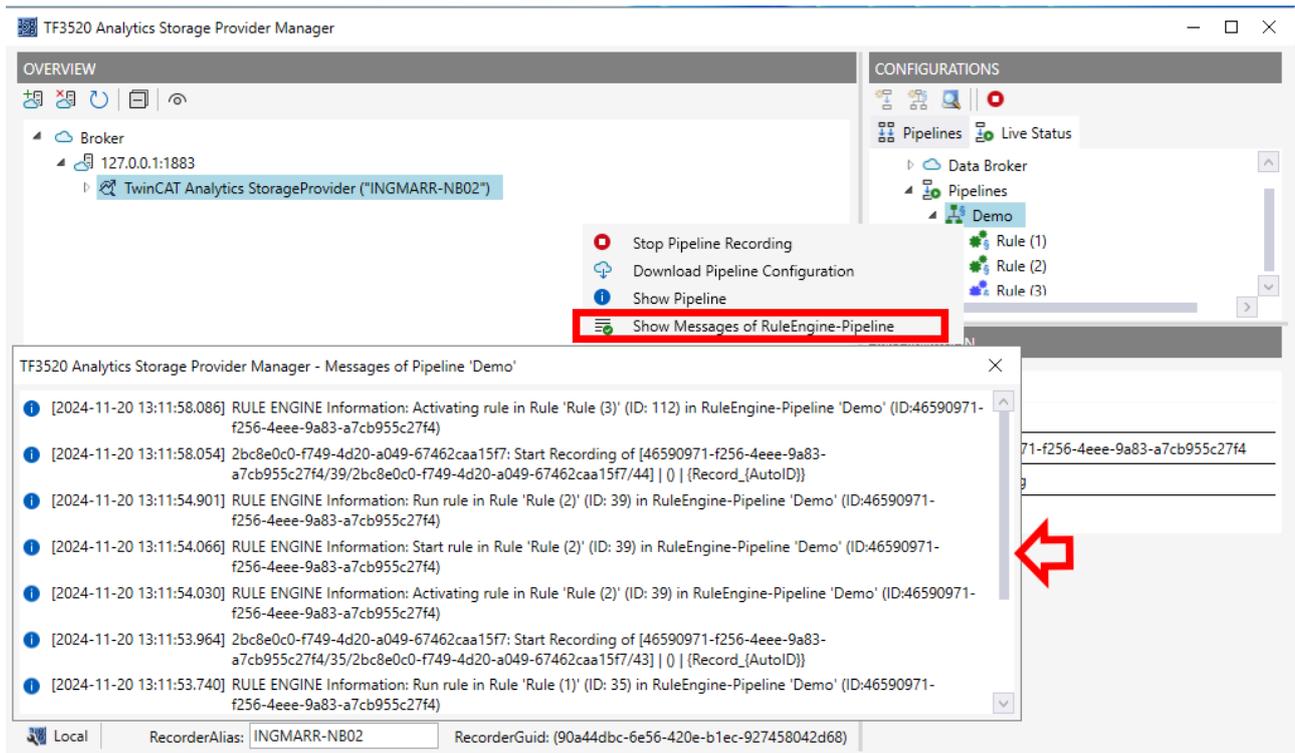
6.2.2.2.4 Rule Zustände

Den Zustand der Rules von der RuleEngine-Pipeline können Sie ebenfalls im Live-Status einsehen. Eine Rule hat folgende Zustände:

	NotInitialized: Die Konfiguration der Rule wurde ausgelesen, jedoch wurde die Rule noch nicht verarbeitet.
	Initializing: Die Rule wird initialisiert. Die notwendigen Sourcen werden in der RuleEngine erzeugt.
	Deactivated: Die Rule ist bereit für die Aktivierung. Zum aktuellen Zeitpunkt läuft sie noch nicht.
	Activating: Die Rule wird aktiviert. Eventuelle Trigger werden gestartet.
	Pending: Die Rule ist aktiviert. Eventuelle Trigger sind gestartet. Die Rule kann jetzt manuell angetriggert werden. StreamingDataRules gehen vom Pending-Zustand direkt in den Starting-Zustand über.
	Starting: Die Rule wird gestartet. Die entsprechenden Verarbeitungsbausteine werden erzeugt und eine Verbindung zu den Datenquellen wird hergestellt.
	Running: Die Rule läuft und verarbeitet Daten.
	Stopping: Die Rule wird gestoppt. Das Stoppen wird beim Stoppen der RuleEngine-Pipeline aufgerufen. BatchDataRules stoppen zudem, wenn alle Daten verarbeitet wurden. Anschließend geht die Rule in den Pending-Zustand über.
	Deactivating: Die Rule wird deaktiviert. Ein Aufstarten mittels Trigger ist nicht mehr möglich.
	Invalid: Die Rule befindet sich in einem ungültigen Zustand und muss neu gestartet werden.
	Error: Die Rule befindet sich in einem Fehlerzustand. Das kann z. B. passieren, wenn eine Datenquelle (Analytics Logger) nicht verfügbar ist. In diesem Fall startet die Rule automatisch wieder auf, sobald die Datenquelle wieder verfügbar ist. Bei allen weiteren Fehlern muss die Rule manuell neu gestartet werden. Das kann über den Storage Provider Manager oder die API bzw. SPS-Bibliothek erfolgen.

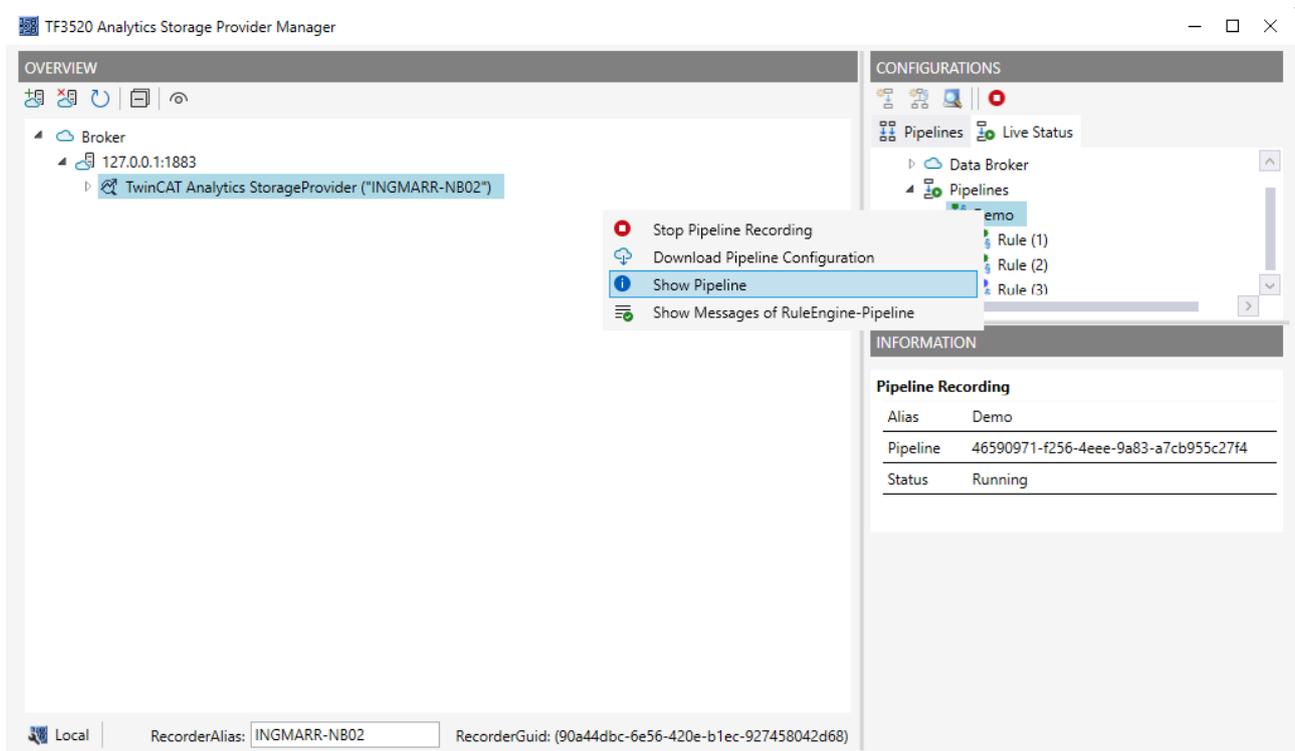
6.2.2.2.5 Messages von RuleEngine-Pipelines

Die Messages von RuleEngine-Pipelines geben genauere Informationen über Fehler sowie Statusinformationen. Die Messages können über das Kontextmenü einer laufenden RuleEngine-Pipeline eingesehen werden. Klicken Sie hierzu auf **Show Messages of RuleEngine-Pipeline**. Ein Fenster mit allen verfügbaren Messages öffnet sich. Befinden sich keine Messages im Fenster sind diese Messages nicht mehr verfügbar. Genauere Informationen befinden sich im Log vom Storage Provider, sofern dieses aktiviert ist.

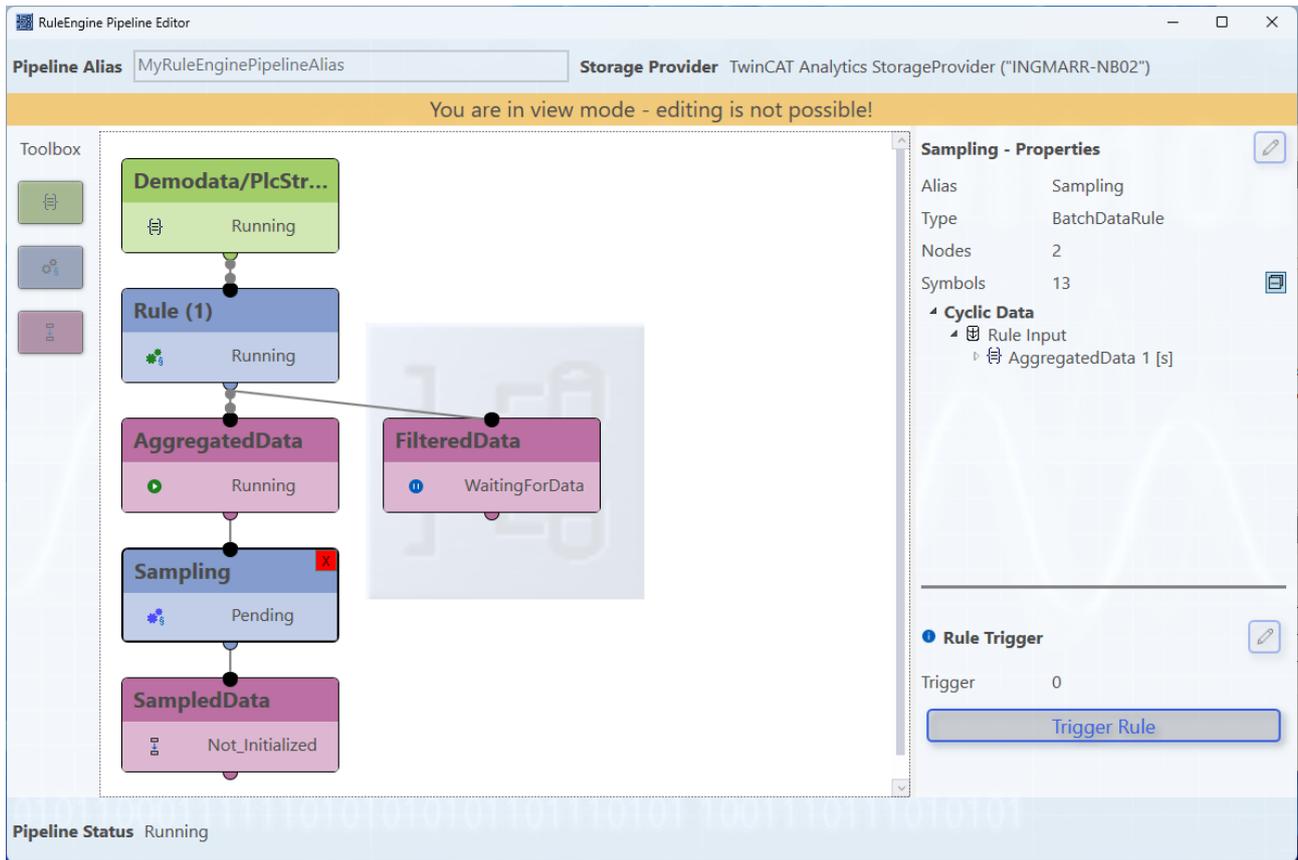


6.2.2.2.6 Triggern und Neustarten von Rules

Das Antriggern oder Neustarten von Rules kann über den RuleEngine-Pipeline-Editor erfolgen. Das Öffnen des Editors muss hierfür über eine laufende RuleEngine-Pipeline (**Kontextmenü > Show Pipeline**) erfolgen.

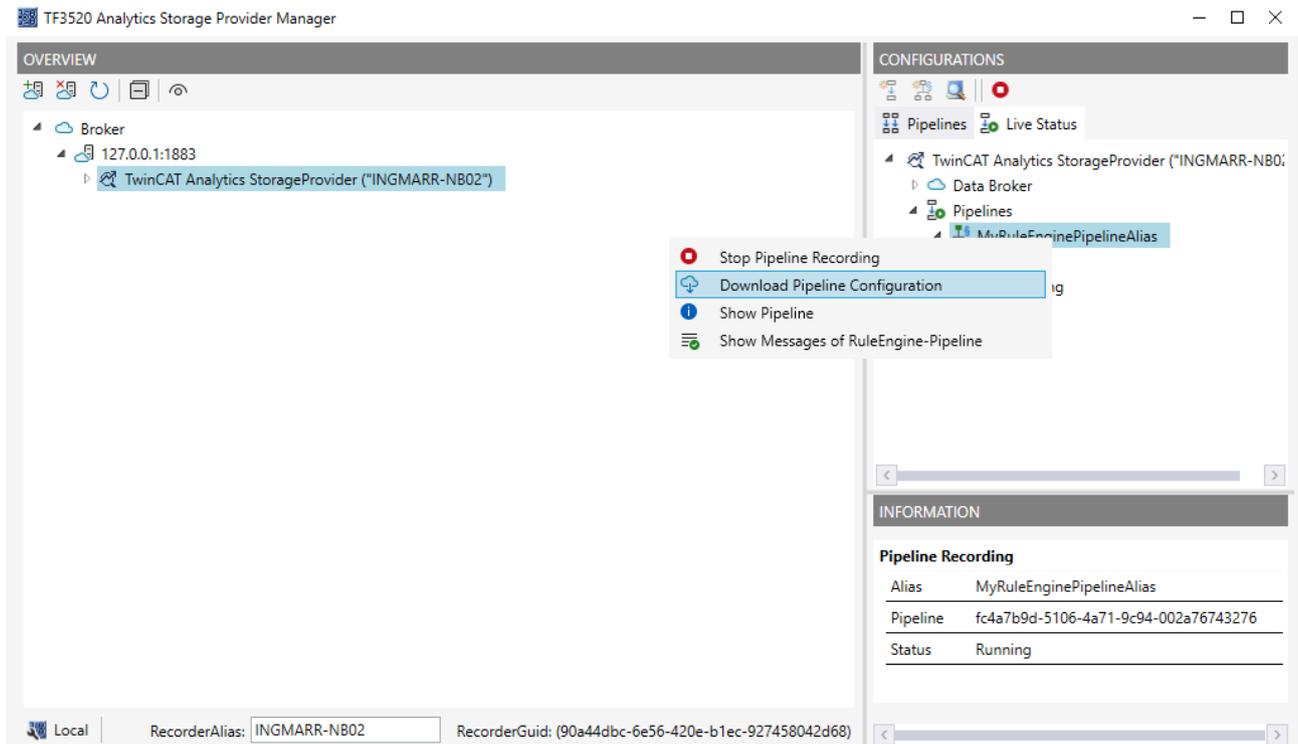


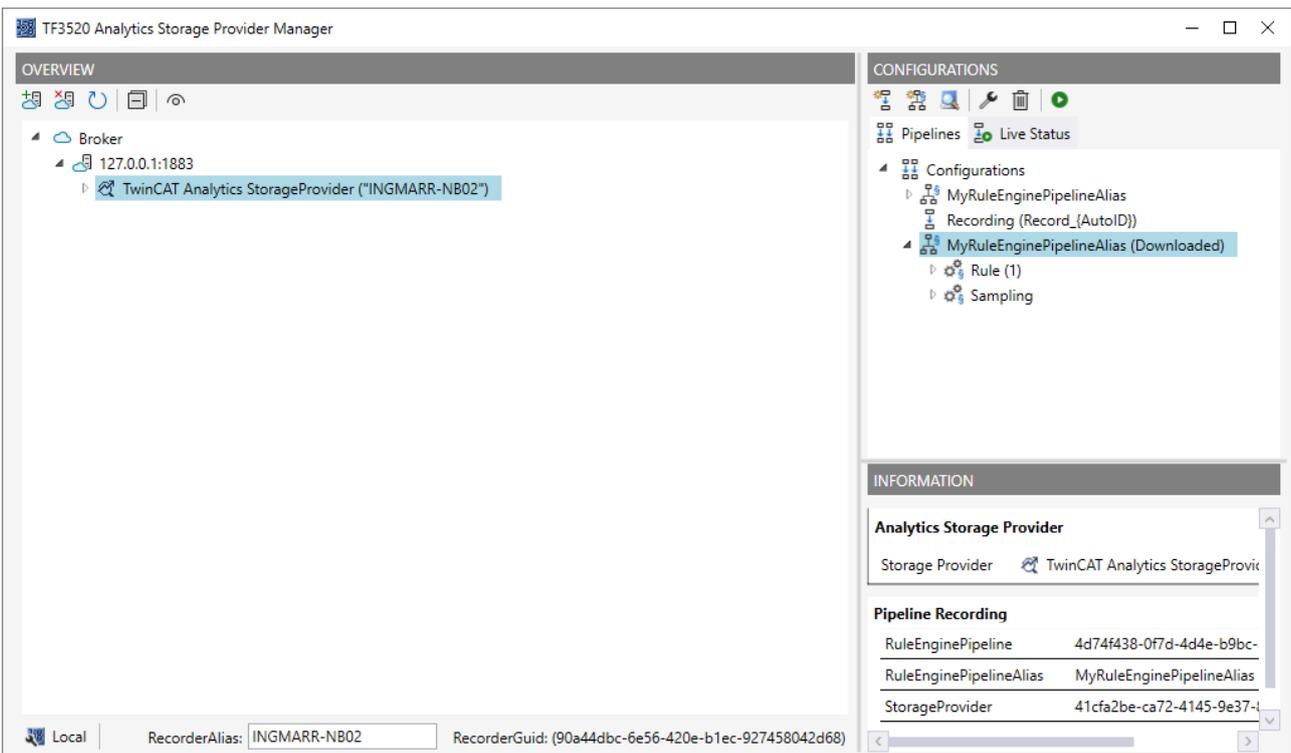
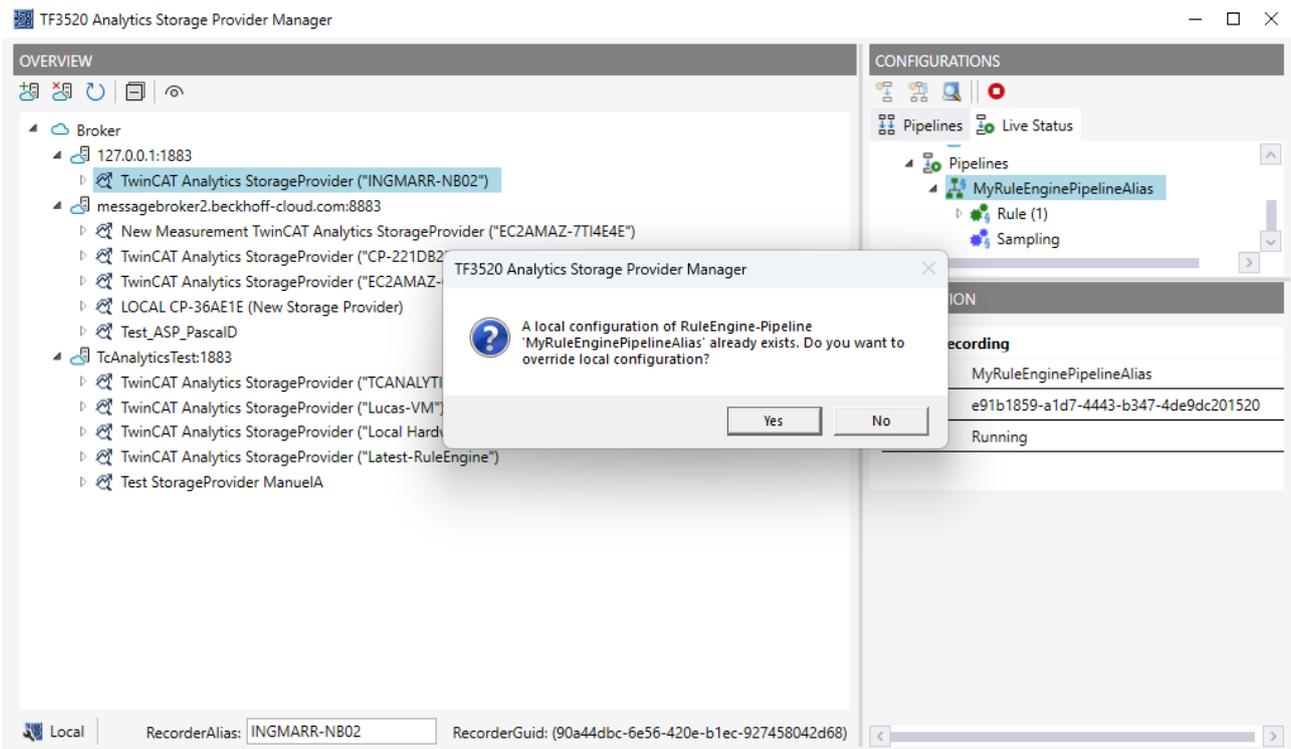
Der RuleEngine-Pipeline-Editor öffnet sich im View-Mode. Dort ist eine Bearbeitung nicht möglich. Im View-Mode kann der Status der einzelnen Elemente eingesehen werden und die entsprechende Rule angetriggert oder neu gestartet werden. Das Triggern ist nur möglich, wenn sich die Rule im Pending-Zustand befindet. Ein Neustart kann nur durchgeführt werden, wenn die Rule in einem Fehlerzustand oder nicht initialisiert ist.



6.2.2.2.7 Download von RuleEngine-Pipelines

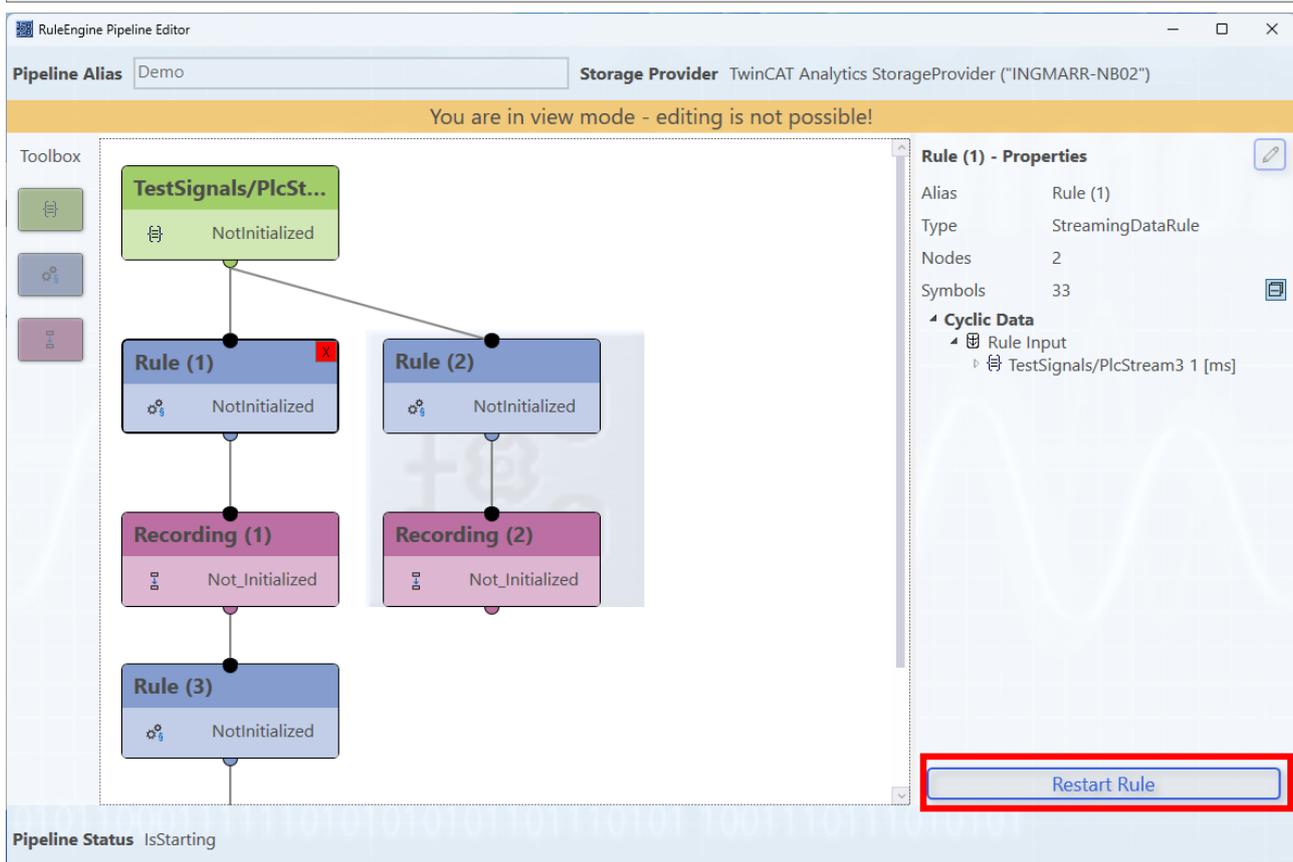
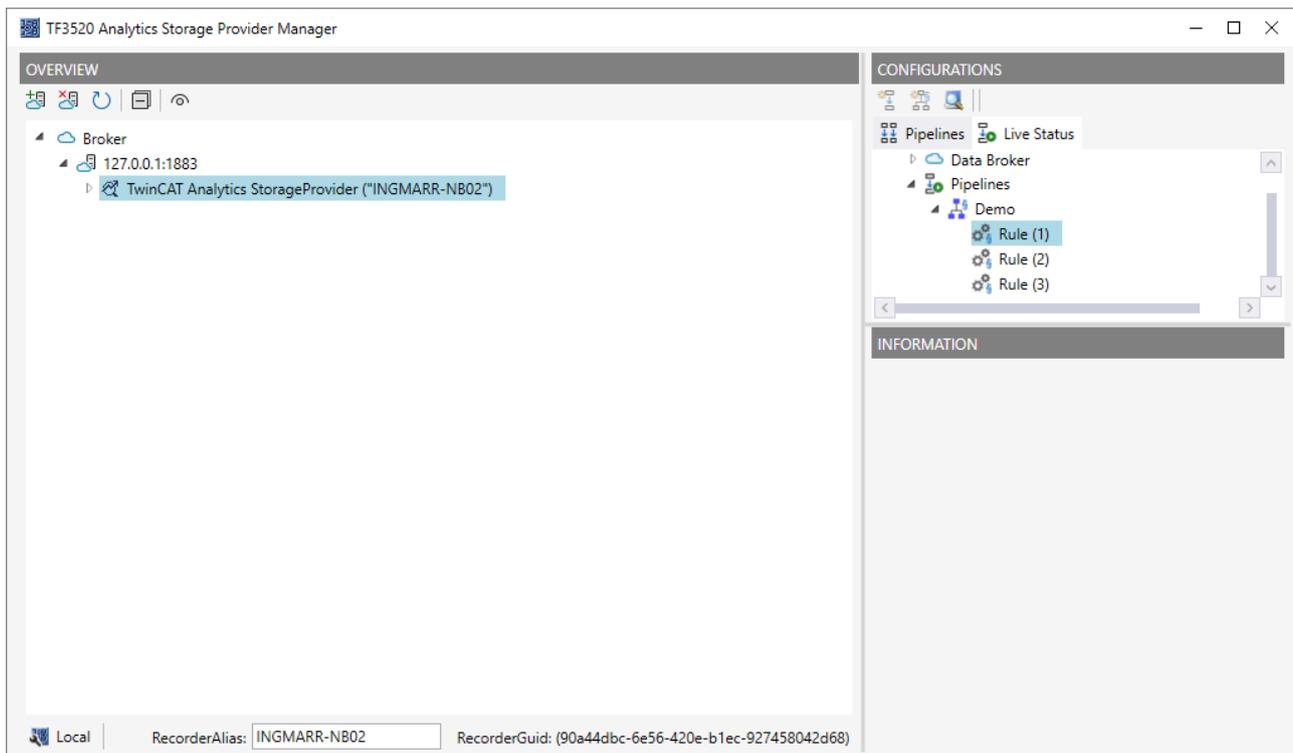
Von einer laufenden RuleEngine-Pipeline kann eine Kopie dieser Pipeline über einen **Rechtsklick > Download** in den lokalen Manager heruntergeladen werden. Liegt dort bereits eine RuleEngine-Pipeline mit der gleichen Pipeline-ID, können Sie auswählen, ob diese überschrieben oder eine Kopie mit einer neuen Pipeline-ID erzeugt werden soll.





6.2.2.2.8 Fehlerfall der RuleEngine

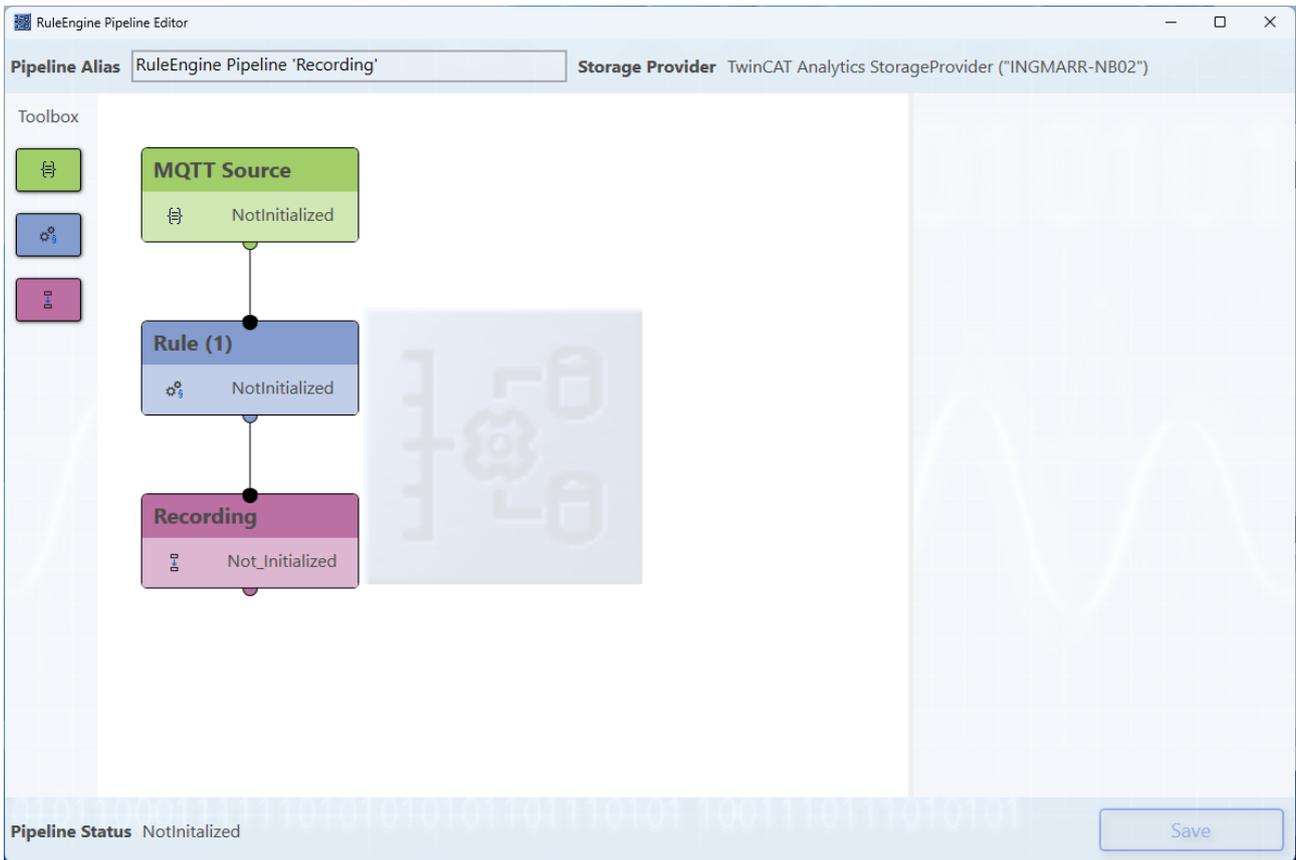
Kommt es zu einem Fehler innerhalb der RuleEngine, welchen diese nicht selbstständig beheben kann, wird die RuleEngine neu gestartet. Alle vorhandenen RuleEngine-Pipelines werden hierfür heruntergefahren. Nach dem Neustart der RuleEngine werden die RuleEngine-Pipelines wieder angelegt. Die darin enthaltenen Rules werden jedoch nicht automatisch neu gestartet. Dies muss manuell [95] über den RuleEngine-Pipeline-Editor erfolgen.



Genauere Beschreibungen zu dem Fehler können Sie aus den Messages des Storage Providers oder dem Storage Provider Log entnehmen.

6.2.2.3 Konvertieren von alten Recordings

Eine vorhandene Pipeline (früheres Recording) kann über einen Rechtsklick **Convert to RuleEngine-Pipeline** in eine RuleEngine überführt werden. Die neue Konfiguration öffnet sich im RuleEngine-Pipeline-Editor und kann dort bearbeitet werden.



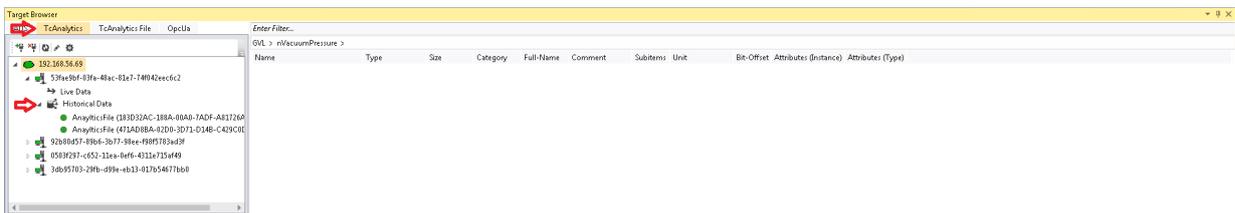
6.3 Arbeit mit historischen Daten

Historische Daten können mit der Analytics Workbench oder dem Analytics Service Tool analysiert werden. Um Ihre aufgezeichneten Daten anzuzeigen, benötigen Sie den TwinCAT Target Browser.

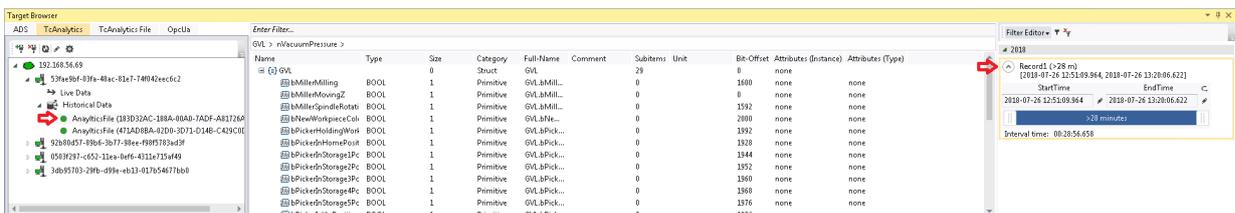
Auswahl der Daten aus dem TwinCAT Target Browser

Die historischen Daten können direkt aus dem Target Browser zu einem Eingang eines Analysealgorithmus gezogen werden.

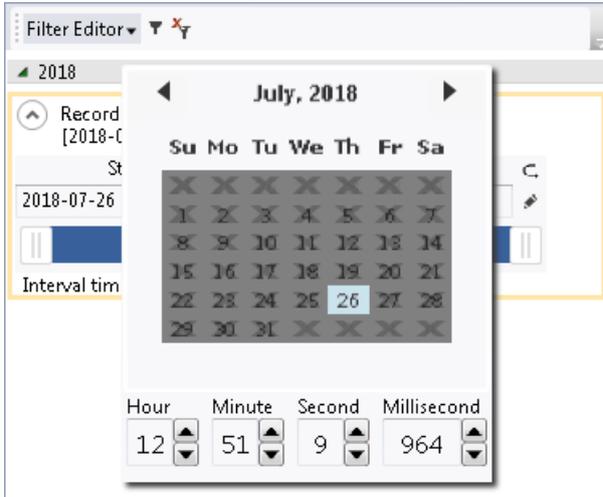
1. Zunächst müssen Sie in der linken Ecke des Target Browser auf **TcAnalytics** klicken. Dort können Sie Ihren konfigurierten Broker sehen, der die Live- und historischen Daten Ihrer verschiedenen Geräte auflistet. Dies sollte wie in der folgenden Abbildung aussehen.



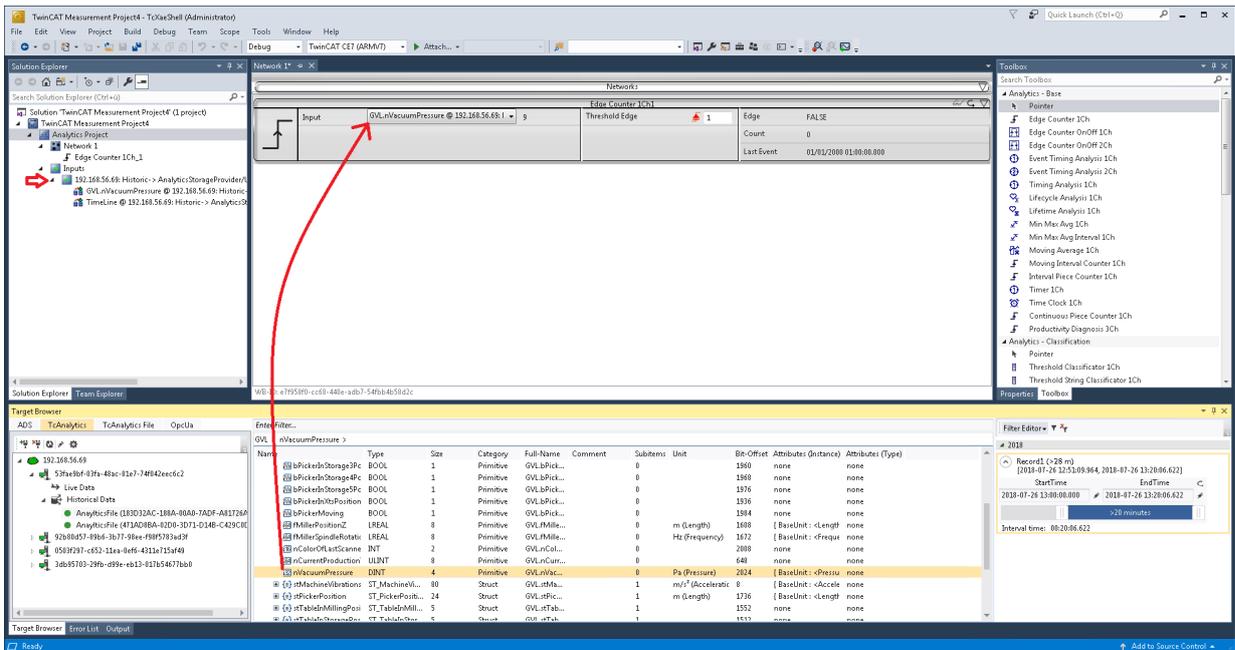
2. Gehen Sie zu dem historischen Stream, den Sie erstellt haben, und wählen Sie die Aufzeichnung aus, die analysiert werden soll. Alle Ihre Aufzeichnungen sind rechts im Fenster **Record** aufgelistet. Standardmäßig wird die letzte Aufzeichnung ausgewählt.



- Wenn Sie live aufzeichnen, wird der Zeitbereich der Aufzeichnung alle paar Sekunden aktualisiert. Standardmäßig wird der gesamte Zeitbereich einer Aufzeichnung verwendet. Außerdem können Sie die Start- und Endzeit bearbeiten, um Ihren gewünschten Datenbereich zu analysieren. Dies kann mit einem Schieberegler, Textfeldern oder in einer grafischen Kalenderansicht erfolgen. Wenn Sie auf das Symbol rechts neben den Textfeldern klicken, wird die Kalenderansicht angezeigt.



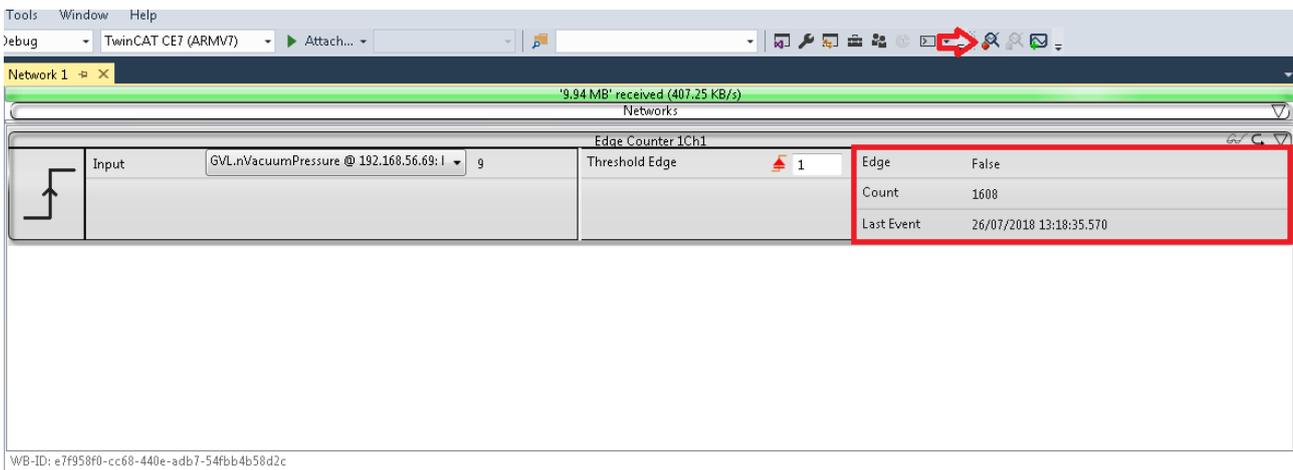
- Nach diesen Schritten können Sie ein Symbol ebenso wie bei den Symbolen der Live-Daten per Drag-and-Drop zu einem Eingang eines Algorithmus ziehen.



⇒ Anschließend wird eine neue Eingangsquelle für Ihren historischen Stream generiert und kann im Projektmappen-Explorer Ihres Visual Studios® angezeigt werden. Als Erstes werden das gezogene Symbol und ein Zeitstempel der aktuellen Gerätezeit unter diesem Stream aufgelistet. Auch neue gezogene Symbole dieses Streams werden dort aufgelistet.

Analyse Ihrer historischen Daten im Analytics-Konfigurator

Um Ihre historischen Daten zu analysieren, drücken Sie auf die Schaltfläche **Start Analytics**. Anders als bei der Analyse von Live-Daten erscheint ein grüner Fortschrittsbalken. Die Geschwindigkeit Ihrer Analyse hängt von der Länge Ihrer Aufzeichnung, der Anzahl und Größe Ihrer Symbole sowie von Ihrer Breitbandgeschwindigkeit zum Broker ab. Die Analyse stoppt automatisch, wenn der Fortschrittsbalken am Ende angelangt ist. Die Ergebnisse bleiben sichtbar.



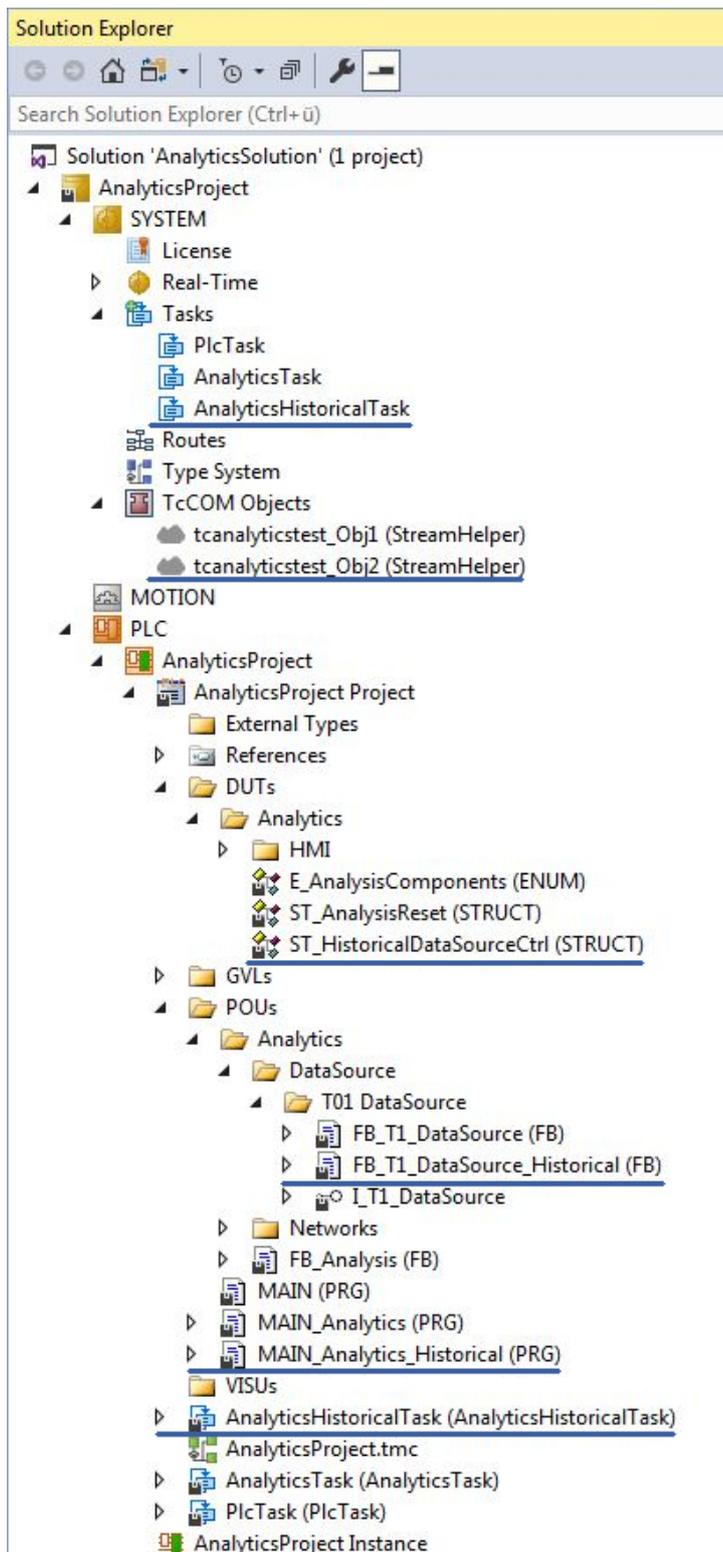
Analyse Ihrer historischen Daten in Ihrer Analytics Runtime

Sie können die Konfiguration mit Ihren historischen Daten einer Analytics Runtime (SPS) bereitstellen. Zusätzlich zu den historischen Daten werden auch die Live-Daten analysiert. So können Sie zwischen ihnen wechseln und verlieren keine Live-Daten durch das Streamen historischer Daten. Der Grund hierfür ist, dass sie in zwei verschiedene Tasks getrennt sind. Der Start der Analyse historischer Daten muss ausgelöst werden.

i Rechenzeit bei historischen Daten

Anders als beim Analytics-Konfigurator dauert die Analyse historischer Daten in der SPS ähnlich lange wie die ursprüngliche Aufzeichnung der Daten. In Abhängigkeit der Paketgröße und der eingestellten Abtastrate kann die Abarbeitung der Daten gegenüber der Aufzeichnung verkürzt werden. Zyklusüberschreitungen aufgrund zu großer Pakete müssen jedoch beachtet werden.

Hauptunterschiede der Ordnerstruktur im erstellten SPS-Projekt:



HINWEIS

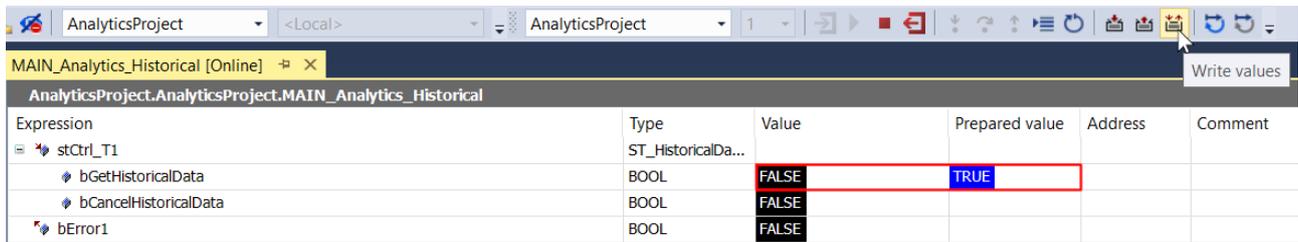
Realisierung der Logik in Ihrer TwinCAT HMI

Die Vorbereitung und das Schreiben von Werten in Ihrer SPS dienen Testzwecken. Es wird empfohlen, diese und andere Logik im SPS-Code bei Bedarf mit Interaktionen von Ihrer TwinCAT HMI-Anwendung zu realisieren.

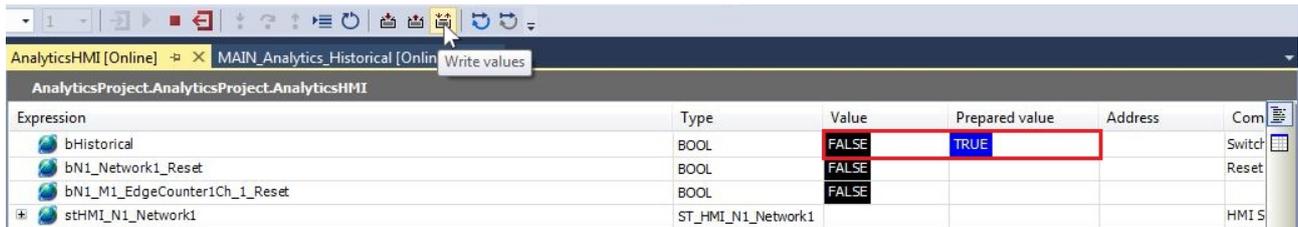
Sie können die Analyse historischer Daten durch Auslösen von **bGetHistoricalData** in **stCtrl_T1** starten.

Der Abbruch erfolgt durch Auslösen von **bCancelHistoricalData**.

Dies kann in der Datei **MAIN_Analytics_Historical** wie in der folgenden Abbildung erfolgen:



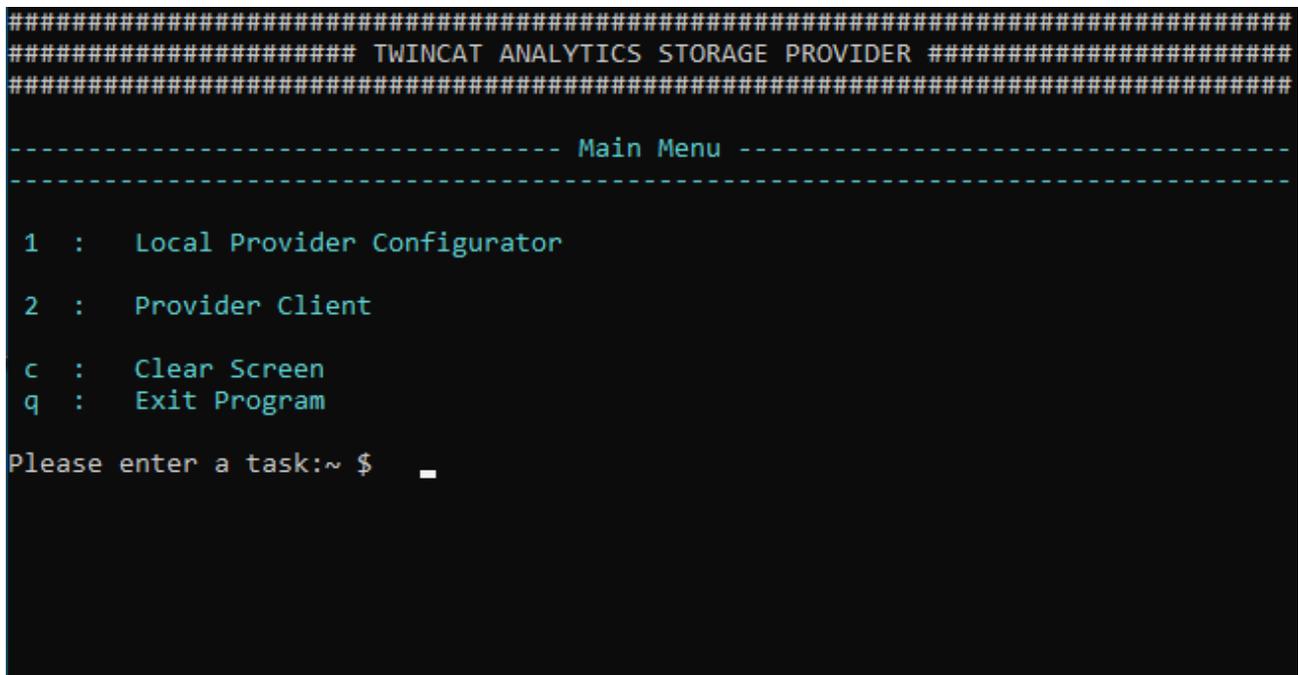
Um für Ihr HMI Dashboard zwischen den Ergebnissen der Live-Daten und historischen Daten zu wechseln, können Sie das Symbol **bHistorical** in der GVL **AnalyticsHMI** setzen. Mit dieser Möglichkeit benötigen Sie keine zusätzlichen Controls, um historische Daten anzuzeigen (natürlich sind auch eigene Controls für historische Daten möglich). Die Analyse der Live-Daten wird nicht durch das Aufrufen einiger historischer Daten unterbrochen. Nach der Anzeige der historischen Daten können Sie zu den aktuellen Live-Ergebnissen zurückwechseln. Dieser Wechsel betrifft nur die Variablen in Ihrer GVL.



6.4 Konsolen- Konfigurator/Client

Der Analytics Storage Provider kann neben dem grafischen Konfigurator und dem Recorder auch über eine Konsole bedient werden. Dadurch kann die Konfiguration und der Zugriff auf einen Storage Provider neben Windows auch unter TwinCAT/BSD durchgeführt werden. Außerdem können mithilfe der Konsolen-Applikation Batch Files zur Steuerung [► 108] des Analytics Storage Providers erzeugt werden.

Nach dem Start des Konsolen-Clients stehen vier Optionen zur Auswahl:



1	Öffnet den Konsolen-Konfigurator [► 104] für den lokalen Analytics Storage Provider
2	Öffnet den Analytics Storage Provider Client [► 105]
c	Leert die Historie des Konsolen-Konfigurators/Clients
q	Schließt den Konsolen-Konfigurator/Client

Durch Eingabe eines der Kennzeichen und bestätigen mit der **[Enter]**-Taste wird die entsprechende Funktion ausgeführt.

6.4.1 Konfigurator

In diesem Menü kann der lokale Analytics Storage Provider konfiguriert werden. Dazu stehen folgende, zusätzliche Eingaben zur Verfügung:

1	Startet einen Dialog zum Konfigurieren des lokalen Analytics Storage Providers
2	Gibt die Konfiguration des lokalen Analytics Storage Providers aus
3	Startet einen Dialog zum Konfigurieren eines Sub Message Broker
4	Gibt alle Sub Message Broker Konfigurationen zurück
5	Festlegen des Master Sub Message Broker (Nur über diesen Broker können Kommandos empfangen werden)
6	Startet einen Dialog zum Konfigurieren eines Storage
7	Gibt alle Storage Konfigurationen zurück
8	Festlegen des Master Storage
10	Startet den Analytics Storage Provider mit der lokalen Konfiguration
11	Stoppt den lokalen Analytics Storage Provider
12	Gibt den Status des Analytics Storage Providers aus
b	Wechselt zurück zum Hauptmenü

Die Konfigurationsparameter sind dieselben wie in dem grafischen Konfigurator des Analytics Storage Providers. Zur Erstellung der Konfiguration werden die Konfigurationsparameter nach der Eingabe:

```

C:\TwinCAT\Functions\TF3520-Analytics-StorageProvider\Client\TwinCAT.Analytics.StoragePr...
#####
##### TWINCAT ANALYTICS STORAGE PROVIDER #####
#####
----- Main Menu Local Provider Configurator -----
-----
1 : Create Local Provider Config
2 : Read Local Provider Config

3 : Create Sub Broker Config
4 : Read Sub Broker Config
5 : Set Master Broker

6 : Create Storage Config
7 : Read Storage Configs
8 : Set Master Storage

10 : Start Local Provider Service
11 : Stop Local Provider Service
12 : Status Local Provider Service

b : Back to Start
c : Clear Screen
q : Exit Program

Please enter a task:~ $

```

Durch die **[ESC]**-Taste kann der Dialog jederzeit abgebrochen und zurück in das Konfigurations-Menü gewechselt werden.

6.4.2 Client

In dem Analytics Storage Provider Client ist es möglich, eine Verbindung zu einem Analytics Storage Provider herzustellen. Dieser muss dabei nicht zwingend lokal auf dem Gerät laufen, sondern kann auch über einen externen MQTT-Broker angesprochen werden. Für die Verwendung des Clients ist es also nicht zwingend notwendig einen Analytics Storage Provider lokal zu konfigurieren. Auf einem verbundenen Analytics Storage Provider können anschließend Aufnahmen, sowie historische Datenstreams an ein konfigurierbares MQTT-Topic gestartet und gestoppt werden.

Dazu stehen folgende Eingaben zur Verfügung:

1	Stellt eine Verbindung mit dem MQTT-Broker aus der lokalen Konfiguration her und selektiert den konfigurierten Analytics Storage Provider.
2	Startet einen Dialog, um eine Verbindung mit einem MQTT-Broker herzustellen .
3	Schließt die Verbindung zu dem aktuellen MQTT-Broker .
10	Stellt die Analytics Storage Provider zur Auswahl bereit, die unter dem MQTT-Broker verfügbar sind .
15	Liest eine Konfigurations-Datei von einem <u>Analytics Storage Provider Recorder</u> , mit den darin konfigurierten Recordings ein .
21	Startet eine RuleEngine-Pipeline.
22	Stoppt eine RuleEngine-Pipeline.
23	Startet eine Rule einer RuleEngine-Pipeline neu.
31	Startet eine Aufnahme.
32	Stoppt eine Aufnahme anhand des Alias und dem MQTT-Topic.
33	Startet einen historischen Datenstream.
34	Stoppt einen historischen Datenstream anhand des Ergebnis-MQTT-Topics.
35	Aktualisiert Streaming-Parameter eines laufenden historischen Datenstreams.
36	Prüft, ob eine Aufnahme aktiv ist.
40	Beendet alle aktiven Aufnahmen.
41	Beendet alle historischen Datenstreams.
50	Löschen von Aufnahmen, die älter als ein bestimmtes Datum sind. Optional kann ein Datenstream angegeben werden.

```
#####
##### TWINCAT ANALYTICS STORAGE PROVIDER #####
#####
RecorderAlias:Console ASP Client ("INGMARR-NB02") - [dac195f2-0000-0000-0000-c0291056d9c1]
----- Main Menu Provider Client -----
1 : Use Local Provider
2 : Connect
3 : Disconnect
10 : Select Provider
15 : Load Config (RecorderSettings)
21 : Start RuleEngine Pipeline
22 : Stop RuleEngine Pipeline
23 : Restart Rule of RuleEngine Pipeline
31 : Start Pipeline
32 : Stop Pipeline
33 : Start Historical Stream
34 : Stop Historical Stream
35 : Set Historical Stream Parameter
36 : Is Recording Active
40 : Cancel All Recordings
41 : Cancel All Historical Streams
50 : Delete Recordings older than specific date of an historical stream (optional)
b : Back to Start
c : Clear Screen
q : Exit Program
Please enter a task:~ $ |
```

Verwendung

Zur Verwendung des Analytics Storage Provider Clients muss zunächst eine Verbindung zu einem MQTT-Broker aufgebaut werden. Durch die Eingabe „2“ wird ein Dialog gestartet, in welchem die bereits konfigurierten MQTT-Broker zur Auswahl gestellt werden. Dort kann durch die Eingabe „0“ ein neuer MQTT-Broker konfiguriert und verbunden werden. Anschließend kann durch die Eingabe „10“ im Client-Hauptmenü ein Analytics Storage Provider ausgewählt werden, welcher unter dem MQTT-Broker verfügbar ist. Alternativ kann durch die Eingabe „1“ im Client-Hauptmenü direkt eine Verbindung mit dem MQTT-Broker und Analytics Storage Provider aus der lokalen Konfigurationsdatei hergestellt werden.

Nach dem erfolgreichen Verbindungsaufbau, wird in der Prompter-Anzeige eine Information über den verbundenen MQTT-Broker und den selektierten Analytics Storage Provider vor der Eingabeaufforderung angezeigt:

```
[TcAnalyticsTest : ASP TcBSD (CX-5AC79C)] Please enter a task:~ $
```

(MQTT-Broker : Analytics Storage Provider)

Zum Starten von Aufnahmen wird mit der Eingabe „31“ ein Dialog gestartet. Dabei besteht die Möglichkeit, eine bereits erstellte Aufnahme zu starten, insofern schon Aufnahmen konfiguriert oder über eine Recorder-Konfigurationsdatei eingelesen wurden. Neben den aufgeführten Aufnahmen kann durch die Eingabe „0“ auch eine neue Aufnahme konfiguriert und gestartet werden:

```

RecorderAlias:Console ASP Client ("MARCT-NB01") - [ffca662c-0000-0000-0000-80d643e8d241] ← 1
----- Main Menu Provider Client -----
1 : Use Local Provider
2 : Connect
3 : Disconnect

10 : Select Provider

20 : Load Config (RecorderSettings)

31 : Start Recording
32 : Stop Recording
33 : Start Historical Stream
34 : Stop Historical Stream
35 : Set Historical Stream Parameter
36 : Is Recording Active

40 : Cancel All Recordings
41 : Cancel All Historical Streams

b : Back to Start
c : Clear Screen
q : Exit Program

[TcAnalyticsTest : ASP TcBSD (CX-5AC79C)] Please enter a task:~ $ 31
Recordings:
(0) - [Create new recording]
(1) - "TcBSD_ASP_Recording" (ASP_Record)
(2) - "TcBSD_ASP_Alias" (TcBSD_Record)
[TcAnalyticsTest : ASP TcBSD (CX-5AC79C)] Please choose id from record config:~ $ 0

[TcAnalyticsTest : ASP TcBSD (CX-5AC79C)] Please enter topic:~ $ TestSignals/StreamFast
[TcAnalyticsTest : ASP TcBSD (CX-5AC79C)] Please enter recording alias:~ $ ASP_Recording
[TcAnalyticsTest : ASP TcBSD (CX-5AC79C)] Please enter recordname:~ $ AnalyticsSP_Record
[TcAnalyticsTest : ASP TcBSD (CX-5AC79C)] Please enter duration in minutes:~ $ 120
[TcAnalyticsTest : ASP TcBSD (CX-5AC79C)] Please enter ringbuffer [None|TimeBased|DataBased]:~ $ None
[TcAnalyticsTest : ASP TcBSD (CX-5AC79C)] Please enter dataFormat [Bin|Json]:~ $ Bin
[TcAnalyticsTest : ASP TcBSD (CX-5AC79C)] Please enter recording mode [All|Subset]:~ $ All
Recording command "ASP_Recording" send to Provider.
Recording "ASP_Recording" is running.

```

Die Konfigurationsparameter entsprechen dabei den bekannten Parametern aus dem grafischen Analytics Storage Provider Recorder. Die Default-Werte können bei Bedarf gelöscht und durch individuelle Eingaben ersetzt werden. Bei dem **recording mode** kann nach der Eingabe „Subset“ eine Untermenge der Symbolik von den Aufnahmedaten definiert werden. Direkt nach der Konfiguration wird ein Kommando zum Start der Aufnahme an den verbundenen Analytics Storage Provider gesendet. Laufende Aufnahmen können über die Eingabe „32“ gestoppt werden. Dabei werden sie durch das MQTT-Topic, von dem die Daten kommen und durch den Aufnahme Alias referenziert. Wenn eine Aufnahme von einem anderen Client gestoppt werden soll, muss zusätzlich die entsprechende **Recorder-Guid** angegeben werden. Die **Recorder Guid** wird gemeinsam mit dem **Recorder-Alias** über den Eingabemöglichkeiten im Client-Hauptmenü angezeigt (rote 1).

Aufnahme-Konfigurationen, die in dem Konsolen Client erstellt wurden, werden nicht persistiert. Nach dem Schließen des Clients ist die Liste der Aufnahme-Konfigurationen also nicht mehr vorhanden. Daher kann das Einlesen von Recorder-Konfigurationsdateien (Eingabe „20“ im Client-Hauptmenü) sehr hilfreich sein. Die Konfigurationsdatei eines Recorders wird unter Windows-Systemen unter dem Pfad `C:\Users***\AppData\Roaming\Beckhoff\TwinCAT Analytics Storage Provider` abgelegt (*** durch den entsprechenden User ersetzen).

Die historisierten Daten des Analytics Storage Providers können über die Eingabe „33“ als Datenstream an ein definierbares Ergebnis-MQTT-Topic übermittelt werden. Dabei wird ebenfalls ein Dialog gestartet, in welchem ein zuvor konfigurierter Datenstream gestartet werden kann. Durch die Eingabe „0“ kann auch ein neuer historischer Datenstream konfiguriert werden:

```

1 : Use Local Provider
2 : Connect
3 : Disconnect

10 : Select Provider

20 : Load Config (RecorderSettings)

31 : Start Recording
32 : Stop Recording
33 : Start Historical Stream
34 : Stop Historical Stream
35 : Set Historical Stream Parameter
36 : Is Recording Active

40 : Cancel All Recordings
41 : Cancel All Historical Streams

b : Back to Start
c : Clear Screen
q : Exit Program

[TcAnalyticsTest : ASP TcBSD (CX-5AC79C)] Please enter a task:~ $ 33
GetHistorical Cmds:
(0) - [Create new GetHistorical cmd]
(1) - "RecordID:29 | Topic:TestSignals/StreamFast"
(2) - "RecordID:39 | Topic:TestSignals/StreamFast"
[TcAnalyticsTest : ASP TcBSD (CX-5AC79C)] Please choose id from cmd:~ $ 0

[TcAnalyticsTest : ASP TcBSD (CX-5AC79C)] Please enter systemID:~ $ cff7975b-b34d-43f7-755d-95cf135f50db
[TcAnalyticsTest : ASP TcBSD (CX-5AC79C)] Please enter topic:~ $ TestSignals/StreamFast
[TcAnalyticsTest : ASP TcBSD (CX-5AC79C)] Please enter symbol layout:~ $ 52a5066f-3c94-d853-f02b-bce62b4a6dea
[TcAnalyticsTest : ASP TcBSD (CX-5AC79C)] Please enter record id:~ $ 1
[TcAnalyticsTest : ASP TcBSD (CX-5AC79C)] Please enter start time in ns:~ $ 13331821883950000
[TcAnalyticsTest : ASP TcBSD (CX-5AC79C)] Please enter end time in ns:~ $ 133318220040054000
[TcAnalyticsTest : ASP TcBSD (CX-5AC79C)] Please enter maxSampleCnt default:~ $ 5000
[TcAnalyticsTest : ASP TcBSD (CX-5AC79C)] Please enter user samplertime default:~ $ -1
[TcAnalyticsTest : ASP TcBSD (CX-5AC79C)] Please enter outputFormat [Bin|Json]:~ $ Bin
[TcAnalyticsTest : ASP TcBSD (CX-5AC79C)] Please enter result topic:~ $ TcBSD_AnalyticsSP_ResultTopic
[TcAnalyticsTest : ASP TcBSD (CX-5AC79C)] Please enter symbol mode [All|Subset]:~ $ All
GetHistoricalStream command for "TcBSD_AnalyticsSP_ResultTopic" send to Provider.

```

Die Parameter definieren dabei eine historisierte Aufnahme, wobei der Parameter **result topic** das MQTT-Topic definiert, an welches die Daten gestreamt werden sollen. Nach der Konfiguration wird automatisch ein Befehl zum Start des Historischen Streams an den Analytics Storage Provider übermittelt.

Durch die Eingabe „35“ in dem Client-Hauptmenü können die Parameter eines aktiven historischen Streams angepasst werden. Der historische Stream wird dabei durch dessen Ergebnis-MQTT-Topic referenziert. Durch die Parameter kann beispielsweise die Geschwindigkeit oder die Paketgröße des Datenstreams bei laufendem Stream angepasst werden. Das Abbrechen eines historischen Datenstreams ist durch die Eingabe „34“ und die Angabe des Ergebnis-MQTT-Topics möglich.

RuleEngine-Pipelines

Auch ein Start von bereits gepublizierten RuleEngine-Pipelines ist möglich. Hierfür muss die Eingabe „21“ erfolgen. Anschließend kann die passende RuleEngine-Pipeline ausgewählt werden. Zudem können RuleEngine-Pipelines gestoppt werden (Eingabe „22“). Auch ein Neustart einzelner Rules (Eingabe „23“) ist möglich.

```

[127.0.0.1 : TwinCAT Analytics StorageProvider ("INGMARR-NB02")] Please enter a task:~ $ 21
RuleEngine Pipelines:
(1) - "MyRuleEnginePipelineAlias" (d00c5366-4cf5-4d4e-a2f6-9dbe759e9dd2)
(2) - "AutostartRuleEnginePipeline" (92bcfd4e-0a05-48a5-b914-80f680d48e92)
[127.0.0.1 : TwinCAT Analytics StorageProvider ("INGMARR-NB02")] Please choose id from pipeline configs:~ $ 1

RuleEngine pipeline command send to Provider.

```

6.4.3 Batch Files zur Steuerung

Mit dem Konsolen-Client können Batch-Files zur Steuerung des Analytics Storage Providers erstellt werden. Dazu werden einige Parameter zur Verfügung gestellt:

-Help / -H / -?	Gibt eine Beschreibung von allen Parametern zurück
-----------------	--

Parameter für die Konfigurationseinstellungen:

-CreateASPConfig	Erstelle eine neue Analytics Storage Provider Einstellungs-XML
-MainTopic <mainTopic>	Analytics Storage Provider Main Topic
-Comment <comment>	Analytics Storage Provider Kommentar
-EventLogTrace <True False>	Trace zum Ereignisprotokoll
-DebugLog <True False>	Zusätzlicher DebugLog
-StorageType <type>	Speichertyp (ANALYTICSFILE , AZURESQL , AZUREBLOB)
-StorageConnString <connString>	Connection String oder Pfad zum Speicher
-TlsType <Tls1.0 Tls1.1 Tls1.2>	Tls Typ (für AzureBlob)
-MaxDuration <duration (sec)>	Maximale Dauer eines TAY-File
-MaxWriteLen <writeLen (bytes)>	Maximale Länge eines Datenpakets

Konfigurationsparameter:

-LocalProvider	Verwende die Verbindungseinstellungen des lokal installierten Analytics Storage Provider
-ConfigFile <path>	Verwende alle Konfigurationen aus der Konfigurationsdatei eines Analytics Storage Provider Recorder Fenster
-ProviderGuid <guid>	Provider des Analytics Storage Providers, der verwendet werden soll
-ConfigCmdID <id>	ID-Nummer der vorkonfigurierten Aufnahme in der Konfigurationsdatei
-ConfigCmdAlias <alias>	Alias von der vorkonfigurierten Aufnahme in der Konfigurationsdatei

Verbindungsparameter:

-Broker /-Host <hostname>	Hostname oder IP-Adresse des verwendeten Brokers
-Port <port>	Port des Brokers (Standardwert: 1883)
-User <username>	Username für die Verbindung
-Password / -Pwd <password>	Passwort für die Verbindung
-CA <path>	Pfad zum CA-Zertifikat für die Verbindung
-Cert <path>	Pfad zum Zertifikat für die Verbindung
-Key_Cert <path>	Pfad zur Schlüssel-Datei für die Verbindung
-Key_Pwd <password>	Passwort für die Schlüsseldatei für die Verbindung

Funktionsparameter:

-StartRecord	Sendet ein StartRecord Kommando
-StopRecord	Sendet ein StopRecord Kommando
-IsRecordingActive	Prüft, ob eine Aufnahme aktuell läuft
-GetHistorical	Sendet ein GetHistoricalData Kommando
-StopHistorical	Sendet ein StopHistoricalData Kommando
-UpdateHistorical	Sendet ein HistoricalUpdate Kommando
-CancelAllRec	Sendet ein Cancel Kommando an alle aktiven Aufnahmen
-CancelAllHist	Sendet ein Cancel Kommando an alle aktiven historischen Datenstreams
-StartPipeline	Sendet ein StartRuleEngine-Pipeline-Kommando
-StopPipeline	Sendet ein StopRuleEngine-Pipeline-Kommando
-RestartRule	Sendet ein RestartRule-Kommando
-DeleteRecordingsOlderThan	Aufnahmen löschen, deren Endzeitpunkt älter als ein vorgegebener Zeitstempel sind. Optional kann zusätzlich das Topic des historischen Streams mit angegeben werden. Es werden nur die aktiven historischen Streams berücksichtigt.

Aufnahme Start/Stop Parameter:

-Alias <alias>	Alias Name der Aufnahme
-RecName <record>	Alias Name der Datensatzes
-Topic <topic>	Topic, das aufgenommen werden soll
-DataFormat <Bin Json>	Datenformat des Live-Datastream
-Duration <seconds>	Dauer der Aufnahme
-Ringbuffer <None TimeBased DataBased>	Ringpuffermodus (Standardwert: Default)
-RinbufferPara <minutes/MB>	Parameter für den Ringpuffer (in Sekunden oder Megabyte)
-Mode <All Subset>	Modus der Aufnahme. Nimmt alle Symbole und eine Untermenge der Symbole auf.
-Symbols / -Sym <Symbol1,Symbol2>	Liste der Symbol-Untermenge als Komma-separierte Liste.
-RecorderGuid <guid>	Guid von dem Analytics Storage Provider Recorder Fenster.
-Storage <guid>	Guid vom Storage wo hineingeschrieben werden soll.
-SubBroker <guid>	Guid vom Sub Broker von dem die Daten aufgezeichnet werden sollen.

Historische Datenstream Start/Stop Parameter:

-SystemID <systemID guid>	System-ID des aufgenommenen Datensatzes.
-Topic <topic>	Topic des aufgenommenen Datensatzes.
-Layout <layout guid>	Layout von dem aufgenommenen Datensatz.
-RecordID <id>	ID von dem Datensatz, der gestreamt werden soll.
-StartTime <time ns>	Startzeitpunkt des Datensatzes, der gestreamt werden soll in Nanosekunden.
-EndTime <time ns>	Endzeitpunkt des Datensatzes, der gestreamt werden soll in Nanosekunden.
-MaxSamples <samples>	Maximale Anzahl Samples (Standardwert: 5000)
-UsrSampleTime <ms>	Abtastrate. (Standardwert: -1; Abtastrate der Aufnahme)
-DataFormat <Bin Json>	Datenformat des Datenstreams.
-ResultTopic <topic>	Ergebnis-MQTT-Topic, auf welches die Daten gestreamt werden.
-Mode <All Subset>	Streaming-Modus. Streamt alle oder eine Untermenge der Symbole.
-Symbols / -Sym <Symbol1, Symbol2>	Liste der Symbol-Untermenge als Komma-separierte Liste.

Historische Datenstream Update Parameter:

-MaxSamples <samples>	Maximale Anzahl Samples (Standardwert: 5000)
-UsrSampleTime <ms>	Abtastrate. (Standardwert: -1; Abtastrate der Aufnahme)
-MaxPackSize <samples>	Maximale Nachrichtengröße in Kilobyte
-SendDuration <ms>	Wartezeit zwischen dem Senden von Nachrichten in Millisekunden.
-ResultTopic <topic>	Ergebnis-MQTT-Topic, auf welches die Daten gestreamt werden.

RuleEngine-Pipeline Parameter:

- PipelineGuid <guid>	Guid der RuleEngine-Pipeline.
- RuleID <id>	ID der Rule innerhalb einer RuleEngine-Pipeline.

Aufnahmen löschen Parameter:

- DateTimeOlderThan <datetime>	Zeitstempel im Format "yyyy-MM-dd hh:mm". Alle Aufnahmen, deren Endzeitpunkt älter als dieser Zeitstempel ist, werden gelöscht.
- HistoricalStreamTopic <topic>	Topic des historischen Streams (optional).

Kommandozeilen Beispiele:

Konfiguration erstellen:

```
TwinCAT.Analytics.StorageProvider.Client
  -CreateASPConfig
    -MainTopic Beckhoff/ASPTest
    -Comment Analytics Storage Provider (Test)
    -EventLogTrace False
    -DebugLog False
    -StorageType ANALYTICSFILE
      -StorageConnString C:\TwinCAT\Functions\TF3520-Analytics-StorageProvider\Storage
      -MaxDuration 120
      -MaxWriteLen 2048
    -Broker 172.17.62.135
      -Port 1883
      -User tcanalytics
      -Pwd 123
```

Aufnahme mit lokalem Analytics Storage Provider starten:

```
TwinCAT.Analytics.StorageProvider.Client
-localprovider
-startrecord
  -alias cmdTest
  -rename cmdRec1
  -topic TestSignals/TestStream
  -dataformat Bin
  -Duration 30
  -mode Subset
  -Symbols Variables.fCosine,Variables.fSine
```

Konfigurationsdatei einer Aufnahme starten:

```
TwinCAT.Analytics.StorageProvider.Client
-ConfigFile "C:
\Users\User\AppData\Roaming\Beckhoff\TwinCAT Analytics Storage Provider\TcAnalyticsStorageProvider_R
ecorder.xml"
-ProviderGuid 76141a7f-e580-4281-99d8-1b8a75ca014d
-startrecord
-ConfigCmdAlias cmdTest
```

Aufnahmestatus prüfen

```
TwinCAT.Analytics.StorageProvider.Client
-Broker 172.17.62.135
-Port 1883
-User tcanalytics
-Pwd 123
-ProviderGuid 76141a7f-e580-4281-99d8-1b8a75ca014d
-IsRecordingActive
  -alias cmdTest
  -recorderGuid a8e171d2-712d-bd8e-da15-7eef28b71ad2
```

Alle Aufnahmen stoppen:

```
TwinCAT.Analytics.StorageProvider.Client
-Broker 172.17.62.135
-Port 1883
-User tcanalytics
-Pwd 123
-ProviderGuid 76141a7f-e580-4281-99d8-1b8a75ca014d
-CancelAllRec
```

Historischen Datenstream starten:

```
TwinCAT.Analytics.StorageProvider.Client
-localprovider
-GetHistorical
  -systemID c29ac2d4-76ce-ff44-4d7f-355ffbcca6bf
  -layout 9a8e171d-712d-bd8e-da15-7eef28b71ad2
  -topic TestSignals/TestStream
  -recordID 1
  -startTime 132696863612730000
  -endTime 13269686417720000
  -maxSamples 5000
  -usrSampleTime -1
  -resultTopic _TestSignals/TestStream/123
  -dataformat Bin
  -mode Subset -symbols Variables.fSine
```

RuleEngine-Pipeline starten:

```
TwinCAT.Analytics.StorageProvider.Client
-localprovider
-StartPipeline
  -PipelineGuid d00c5366-4cf5-4d4e-a2f6-9dbe759e9dd2
```

RuleEngine-Pipeline stoppen:

```
TwinCAT.Analytics.StorageProvider.Client
-localprovider
-StopPipeline
  -PipelineGuid d00c5366-4cf5-4d4e-a2f6-9dbe759e9dd2
```

Spezielle Rule einer RuleEngine-Pipeline starten:

```
TwinCAT.Analytics.StorageProvider.Client
  -localprovider
  -RestartRule
    -PipelineGuid d00c5366-4cf5-4d4e-a2f6-9dbe759e9dd2
    -RuleID 2
```

Alte Aufnahmen löschen:

```
TwinCAT.Analytics.StorageProvider.Client
  -localprovider
  -DeleteRecordingsOlderThan
    -DateTimeOlderThan yyyy-MM-dd 00:00
    - HistoricalStreamTopic Beckhoff /TcAnalyticsStorageProvider/41cfa2be-
ca72-4145-9e37-875851502aa6/Historical/Stream_65
```

7 SPS-API

7.1 Funktionsbausteine

7.1.1 Topic-Architektur

7.1.1.1 Commands

7.1.1.1.1 T_ALY_SPCancel_Cmd

T_ALY_SPCancel_Cmd	
eCancelType	<i>E_CancelType</i>
arrParameter	<i>ARRAY [0..99] OF T_MaxString</i>

Syntax

Definition:

```
FUNCTION_BLOCK T_ALY_SPCancel_Cmd EXTENDS T_ALY_JsonPayload
VAR_INPUT
    eCancelType : E_CancelType;
    arrParameter : ARRAY [0..99] OF T_MaxString;
END_VAR
```

Vererbungshierarchie

[T_ALY_JsonPayload \[► 134\]](#)

T_ALY_SPCancel_Cmd

🔍 Eingänge

Name	Typ	Beschreibung
eCancelType	E_CancelType [► 146]	
arrParameter	ARRAY [0..99] OF T_MaxString	

🔍 Methoden

Name	Definitionsart	Beschreibung
Reset		Zurücksetzen aller Werte im Nutzlast-FB.
Init_JsonValue	Vererbt von T_ALY_JsonPayload [► 134]	Initialisierung des FB mit JSON-Objekt.
Init_String	Vererbt von T_ALY_JsonPayload [► 134]	Initialisierung des FB mit JSON-String.
GetJsonLength	Vererbt von T_ALY_JsonPayload [► 134]	Abrufen der Länge der JSON-Nutzlast.
GetJsonString	Vererbt von T_ALY_JsonPayload [► 134]	Abrufen der JSON-Nutzlast als String.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.25	PC oder CX (x64, x86, Arm®)	Tc3_AnalyticsStorageProvider

7.1.1.1.2 T_ALY_SPGetHistorical_Cmd

T_ALY_SPGetHistorical_Cmd	
—sTopic	T_MaxString
—sLayout	GUID
—eMode	E_SymbolMode
—eOutputFormat	E_RawDataFormat
—nMaxSampleCount	UDINT
—nUserSampleTime	DINT
—nRecordID	DINT
—nStartTimestamp	LINT
—nEndTimestamp	LINT
—sResultTopic	T_MaxString
—arrSymbols	ARRAY [0..255] OF T_ALY_Symbol

Syntax

Definition:

```
FUNCTION_BLOCK T_ALY_SPGetHistorical_Cmd EXTENDS T_ALY_JsonPayload
VAR_INPUT
    sTopic      : T_MaxString;
    sLayout     : GUID;
    eMode       : E_SymbolMode := E_SymbolMode.All;
    eOutputFormat : E_RawDataFormat := E_RawDataFormat.Bin;
    nMaxSampleCount : UDINT := 3000;
    nUserSampleTime : DINT := -1;
    nRecordID    : DINT;
    nStartTimestamp : LINT;
    nEndTimestamp  : LINT;
    sResultTopic  : T_MaxString;
    arrSymbol    : ARRAY [0..255] OF T_ALY_Symbol;
END_VAR
```

Vererbungshierarchie

[T_ALY_JsonPayload \[► 134\]](#)

T_ALY_SPGetHistorical_Cmd

Eingänge

Name	Typ	Beschreibung
sTopic	T_MaxString	Topic-Name des aufgezeichneten Live-Streams.
sLayout	GUID	Layout-GUID der Aufzeichnung.
eMode	E_SymbolMode  150	Abrufen aller Symbole oder nur einer Teilmenge.
eOutputFormat	E_RawDataFormat  148	Format der zurückgegebenen Daten (derzeit wird nur „Bin“ unterstützt).
nMaxSampleCount	UDINT	Maximale Anzahl von Samples in einem Nutzlastpaket.
nUserSampleTime	DINT	Samplezeit in Millisekunden des zurückgegebenen Streams. (-1 verwendet die aufgezeichnete Samplezeit.)
nRecordID	DINT	Nummer der Aufzeichnung.
nStartTimestamp	LINT	Startzeit
nEndTimestamp	LINT	Endzeit
sResultTopic	T_MaxString	Topic-Name des Ergebnis-Streams.
arrSymbol	ARRAY [0..255] OF T_ALY_Symbol  132	Wenn der SymbolMode Subset ist, wird nur die Liste dieser Symbole zurückgegeben.

Methoden

Name	Definitionsart	Beschreibung
Reset		Zurücksetzen aller Werte im Nutzlast-FB.
Init_JsonValue	Vererbt von T_ALY_JsonPayload  134	Initialisierung des FB mit JSON-Objekt.
Init_String	Vererbt von T_ALY_JsonPayload  134	Initialisierung des FB mit JSON-String.
GetJsonLength	Vererbt von T_ALY_JsonPayload  134	Abrufen der Länge der JSON-Nutzlast.
GetJsonString	Vererbt von T_ALY_JsonPayload  134	Abrufen der JSON-Nutzlast als String.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.25	PC oder CX (x64, x86, Arm®)	Tc3_AnalyticsStorageProvider

7.1.1.1.3 T_ALY_SPReadStreamRecords_Cmd



Syntax

Definition:

```

FUNCTION_BLOCK T_ALY_SPReadStreamRecords_Cmd EXTENDS T_ALY_JsonPayload
VAR_INPUT
    sStreamTopic : STRING(255);
    sStreamSystemID : GUID;
    sStreamLayout : GUID;
    nRecordStartIndex : DINT;
    nMaxRecordCount : DINT;
    sResultTopic : T_MaxString;
END_VAR
    
```

Vererbungshierarchie

T_ALY_JsonPayload [▶ 134]

T_ALY_SPReadStreamRecords_Cmd

Eingänge

Name	Typ	Beschreibung
sStreamTopic	STRING(255)	Topic-Name des aufgezeichneten Live-Streams.
sStreamSystemID	GUID	SystemID des Target Systems von wo der Live-Stream gesendet wurde
sStreamLayout	GUID	Layout-GUID der Aufzeichnung
nRecordStartIndex	DINT	Startindex der ersten Aufzeichnung die gelesen werden soll.
nMaxRecordCount	DINT	Gesamtanzahl der Aufzeichnungen, die gelesen werden soll.
sResultTopic	T_MaxString	Topic-Name des Ergebnis-Streams

Methoden

Name	Definitionort	Beschreibung
Reset		Zurücksetzen aller Werte im Nutzlast-FB.
Init_JsonValue	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-Objekt.
Init_String	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-String.
GetJsonLength	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der Länge der JSON-Nutzlast.
GetJsonString	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der JSON-Nutzlast als String.
Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.25	PC oder CX (x64, x86, Arm®)	Tc3_AnalyticsStorageProvider

7.1.1.1.4 T_ALY_SPRecordData_Cmd

T_ALY_SPRecordData_Cmd	
—sAlias	<i>T_MaxString</i>
—sRecordName	<i>T_MaxString</i>
—eRecording	<i>E_RecordMode</i>
—sRecorder	<i>GUID</i>
—sRecorderAlias	<i>T_MaxString</i>
—sTopic	<i>T_MaxString</i>
—eDataFormat	<i>E_RawDataFormat</i>
—nDuration	<i>DINT</i>
—eRingBufferMode	<i>E_RingBufferMode</i>
—nRingBufferParameter	<i>DINT</i>
—eMode	<i>E_SymbolMode</i>
—sSymbolLayout	<i>GUID</i>
—arrSymbols	<i>ARRAY [0..255] OF T_ALY_Symbol</i>

Syntax

Definition:

```
FUNCTION_BLOCK T_ALY_SPRecordData_Cmd EXTENDS T_ALY_JsonPayload
VAR_INPUT
  sAlias : T_MaxString;
  sRecordName : T_MaxString;
  eRecording : E_RecordMode;
  sRecorder : GUID;
  sRecorderAlias : T_MaxString;
  sTopic : T_MaxString;
  eDataFormat : E_RawDataFormat;
  nDuration : DINT;
  eRingBufferMode : E_RingBufferMode;
  nRingBufferParameter : DINT;
  eMode : E_SymbolMode;
  sSymbolLayout : GUID;
  arrSymbols : ARRAY [0..255] OF T_ALY_Symbol;
END_VAR
```

Vererbungshierarchie

[T_ALY_JsonPayload](#) [\[▶ 134\]](#)

T_ALY_SPRecordData_Cmd

 Eingänge

Name	Typ	Beschreibung
sAlias	T_MaxString	Alias-Name für die Aufzeichnung.
sRecordName	T_MaxString	Name für diese Aufzeichnung.
eRecording	E_RecordMode [▶ 148]	Start oder Stopp der Aufzeichnung.
sRecorder	GUID	Individuelle GUID des Recorders.
sRecorderAlias	T_MaxString	Alias-Name für den Recorder.
sTopic	T_MaxString	Topic-Name des Live-Streams.
eDataFormat	E_RawDataFormat [▶ 148]	Speichern des Datenformats (derzeit wird nur das Binärformat unterstützt).
nDuration	DINT	Dauer der Aufzeichnung in Minuten. (-1 unbegrenzt)
eRingBufferMode	E_RingBufferMode [▶ 149]	Ringpuffermodus
nRingBufferParameter	DINT	TimeBased => Parameter in Minuten. DataBased => Parameter in Megabyte.
eMode	E_SymbolMode [▶ 150]	Aufzeichnen aller Symbole oder nur einer Teilmenge.
sSymbolLayout	GUID	
arrSymbols	ARRAY [0..255] OF T_ALY_Symbol [▶ 132]	Wenn der SymbolMode Subset ist, wird nur die Liste dieser Symbole aufgezeichnet.

 Methoden

Name	Definitionsort	Beschreibung
Reset		Zurücksetzen aller Werte im Nutzlast-FB.
Init_JsonValue	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-Objekt.
Init_String	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-String.
GetJsonLength	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der Länge der JSON-Nutzlast.
GetJsonString	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der JSON-Nutzlast als String.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.25	PC oder CX (x64, x86, Arm®)	Tc3_AnalyticsStorageProvider

7.1.1.1.5 T_ALY_SPReloadHistoricalStreams_Cmd

```

T_ALY_SPReloadHistoricalStreams_Cmd
---eReloadType E_ReloadType
---arrParameter ARRAY[0..9] OF ARRAY[0..1] OF T_MaxString
    
```

Syntax

Definition:

```
FUNCTION_BLOCK T_ALY_SPReloadHistoricalStreams_Cmd EXTENDS T_ALY_JsonPayload
VAR_INPUT
    eReloadType : E_ReloadType;
    arrParameter : ARRAY [0..9] OF ARRAY [0..1] OF T_MaxString;
END_VAR
```

Vererbungshierarchie

[T_ALY_JsonPayload \[► 134\]](#)

T_ALY_SPReloadHistoricalStreams_Cmd

📌 Eingänge

Name	Typ	Beschreibung
eReloadType	E_ReloadType [► 149]	Auswahl des Aktualisierungsmodus
arrParameter	ARRAY [0..9] OF ARRAY [0..1] OF T_MaxString	Zusatzparameter

🔗 Methoden

Name	Definitionsart	Beschreibung
Reset		Zurücksetzen aller Werte im Nutzlast-FB.
Init_JsonValue	Vererbt von T_ALY_JsonPayload [► 134]	Initialisierung des FB mit JSON-Objekt.
Init_String	Vererbt von T_ALY_JsonPayload [► 134]	Initialisierung des FB mit JSON-String.
GetJsonLength	Vererbt von T_ALY_JsonPayload [► 134]	Abrufen der Länge der JSON-Nutzlast.
GetJsonString	Vererbt von T_ALY_JsonPayload [► 134]	Abrufen der JSON-Nutzlast als String.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.25	PC oder CX (x64, x86, Arm®)	Tc3_AnalyticsStorageProvider

7.1.1.1.6 T_ALY_SPSetGetHistoricalDataState_Cmd

T_ALY_SPSetGetHistoricalDataState_Cmd	
—	sResultTopic <i>T_MaxString</i>
—	eState <i>E_SetGetHistoricalDataState</i>
—	nSendDuration_ms <i>DINT</i>
—	nRestartTimestamp <i>LINT</i>
—	nMaxSampleCount <i>UDINT</i>
—	nMaxPackageSize_KB <i>DINT</i>
—	nUserSampleTime <i>LINT</i>

Syntax

Definition:

```
FUNCTION_BLOCK T_ALY_SPSetGetHistoricalDataState_Cmd EXTENDS T_ALY_JsonPayload
VAR_INPUT
    sResultTopic : T_MaxString;
    eState : E_SetGetHistoricalDataState;
    nSendDuration_ms : DINT;
    nRestartTimestamp : LINT;
    nMaxSampleCount : UDINT;
    nMaxPackageSize_KB: DINT;
    nUserSampleTime : LINT;
END_VAR
```

Vererbungshierarchie

T_ALY_JsonPayload [[▶ 134](#)]

T_ALY_SPSetGetHistoricalDataState_Cmd

 **Eingänge**

Name	Typ	Beschreibung
sResultTopic	T_MaxString	Topic-Name des Ergebnis-Streams (wird wie ein Handle verwendet).
eState	E_SetGetHistoricalDataState ▶ 149	Status des Historischen-Streams
nSendDuration_ms	DINT	Wartezeit zwischen dem versenden der einzelnen Paketen
nRestartTimestamp	LINT	Zeitstempel an dem der Ergebnis-Stream fortgeführt wird.
nMaxSampleCount	UDINT	Maximale Anzahl an Einträgen in einem Paket
nMaxPackageSize_KB	DINT	Maximal Größe eines Pakets
nUserSampleTime	LINT	Samplezeit in Millisekunden des zurückgegebenen Streams (-1 verwendet die aufgezeichnete Samplezeit).

 **Methoden**

Name	Definitionsart	Beschreibung
Reset		Zurücksetzen aller Werte im Nutzlast-FB.
Init_JsonValue	Vererbt von T_ALY_JsonPayload ▶ 134	Initialisierung des FB mit JSON-Objekt.
Init_String	Vererbt von T_ALY_JsonPayload ▶ 134	Initialisierung des FB mit JSON-String.
GetJsonLength	Vererbt von T_ALY_JsonPayload ▶ 134	Abrufen der Länge der JSON-Nutzlast.
GetJsonString	Vererbt von T_ALY_JsonPayload ▶ 134	Abrufen der JSON-Nutzlast als String.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.25	PC oder CX (x64, x86, Arm®)	Tc3_AnalyticsStorageProvider

7.1.1.1.7 T_ALY_SPStorageCtrl_Cmd



Syntax

Definition:

```

FUNCTION_BLOCK T_ALY_SPStorageCtrl_Cmd EXTENDS T_ALY_JsonPayload
VAR_INPUT
    eCtrlMode : E_ControlMode;
    sStorageGuid : GUID;
END_VAR

```

Vererbungshierarchie

[T_ALY_JsonPayload \[▶ 134\]](#)

T_ALY_SPRecordData_Cmd

🔌 Eingänge

Name	Typ	Beschreibung
eCtrlMode	E_ControlMode [▶ 147]	Start, Stopp, etc des Storage.
sStorageGuid	GUID	Individuelle GUID des Storage.

🔧 Methoden

Name	Definitionsort	Beschreibung
Reset		Zurücksetzen aller Werte im Nutzlast-FB.
Init_JsonValue	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-Objekt.
Init_String	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-String.
GetJsonLength	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der Länge der JSON-Nutzlast.
GetJsonString	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der JSON-Nutzlast als String.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.25	PC oder CX (x64, x86, Arm®)	Tc3_AnalyticsStorageProvider

7.1.1.1.8 T_ALY_SPRuleEnginePipeline_Cmd



Syntax

Definition:

```
FUNCTION_BLOCK T_ALY_SPRecordData_Cmd EXTENDS T_ALY_JsonPayload
VAR_INPUT
    sRuleEnginePipeline : GUID;
    eCmdType : E_PipelineCmdType;
    sRecorder : GUID;
    sRecorderAlias : T_MaxString;
END_VAR
```

Vererbungshierarchie

T_ALY_JsonPayload [▶ 134]

T_ALY_SPRuleEnginePipeline_Cmd

Eingänge

Name	Typ	Beschreibung
sRuleEnginePipeline	GUID	Individuelle GUID der RuleEngine Pipeline.
eCmdType	E_PipelineCmdType [▶ 151]	Start oder Stopp der Aufzeichnung.
sRecorder	GUID	Individuelle GUID des Recorders.
sRecorderAlias	T_MaxString	Alias-Name für den Recorder.

Methoden

Name	Definitionsart	Beschreibung
Reset		Zurücksetzen aller Werte im Nutzlast-FB.
Init_JsonValue	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-Objekt.
Init_String	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-String.
GetJsonLength	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der Länge der JSON-Nutzlast.
GetJsonString	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der JSON-Nutzlast als String.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.25	PC oder CX (x64, x86, Arm®)	Tc3_AnalyticsStorageProvider

7.1.1.2 Descriptions

7.1.1.2.1 T_ALY_HistoricalStream_Desc



Syntax

Definition:

```

FUNCTION_BLOCK T_ALY_HistoricalStream_Desc EXTENDS T_ALY_JsonPayload
VAR_OUTPUT
    sSource : STRING(255);
    sStreamTopic : STRING(255);
    sStreamAlias : STRING(255);
    sStreamSystemID : GUID;
    sLayout : GUID;
    nCycleTime: UDINT;
    nDataSize : UDINT;
    arrRecords : ARRAY [0..cMaxRecordCount] OF T_RecordTimestamps;
END_VAR

```

Vererbungshierarchie

T_ALY_JsonPayload [\[► 134\]](#)

T_ALY_HistoricalStream_Desc

Ausgänge

Name	Typ	Beschreibung
sSource	STRING(255)	Name der Datenquelle
sStreamTopic	STRING(255)	Topic-Name des aufgezeichneten Streams
sStreamAlias	STRING(255)	Alias-Name des Streams
sStreamSystemID	GUID	SystemID-GUID des Streams
sLayout	GUID	Layout-GUID der Aufzeichnung
nCycleTime	UDINT	Zykluszeit der Aufzeichnung
nDataSize	UDINT	Datengröße eines Eintrags der Aufzeichnung
arrRecords	ARRAY [0..cMaxRecordCount] OF T_RecordTimestamps [► 133]	Zeitstempel der verschiedenen Aufzeichnungen

Methoden

Name	Definitionsart	Beschreibung
Reset		Zurücksetzen aller Werte im Nutzlast-FB.
Init_JsonValue	Vererbt von T_ALY_JsonPayload [► 134]	Initialisierung des FB mit JSON-Objekt.
Init_String	Vererbt von T_ALY_JsonPayload [► 134]	Initialisierung des FB mit JSON-String.
GetJsonLength	Vererbt von T_ALY_JsonPayload [► 134]	Abrufen der Länge der JSON-Nutzlast.
GetJsonString	Vererbt von T_ALY_JsonPayload [► 134]	Abrufen der JSON-Nutzlast als String.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.25	PC oder CX (x64, x86, Arm®)	Tc3_AnalyticsStorageProvider

7.1.1.2.2 T_ALY_SPIInstance_Desc

T_ALY_SPIInstance_Desc	
DATE_AND_TIME	dtTimestamp
BOOL	bOnline
STRING	sName
STRING	sVersion
T_ALY_SPIInstanceInfo	stInfo

Syntax

Definition:

```
FUNCTION_BLOCK T_ALY_SPInstance_Desc EXTENDS T_ALY_JsonPayload
VAR_OUTPUT
    dtTimestamp : DATE_AND_TIME;
    bOnline     : BOOL;
    sName       : STRING;
    sVersion    : STRING;
    stInfo      : T_ALY_SPInstanceInfo;
END_VAR
```

Vererbungshierarchie

T_ALY_JsonPayload [[▶ 134](#)]

T_ALY_SPInstance_Desc

 **Ausgänge**

Name	Typ	Beschreibung
dtTimestamp	DATE_AND_TIME	Startzeitpunkt des Storage Provider Service
bOnline	BOOL	Signalisiert, ob der Service Online ist.
sName	STRING	IoT Device Name „TwinCAT Analytics Storage Provider“
sVersion	STRING	Version des Storage Provider
stInfo	T_ALY_SPInstanceInfo ▶ 129	Detaillierte Beschreibung des Storage Providers

 **Methoden**

Name	Definitionsart	Beschreibung
Reset		Zurücksetzen aller Werte im Nutzlast-FB.
Init_JsonValue	Vererbt von T_ALY_JsonPayload ▶ 134	Initialisierung des FB mit JSON-Objekt.
Init_String	Vererbt von T_ALY_JsonPayload ▶ 134	Initialisierung des FB mit JSON-String.
GetJsonLength	Vererbt von T_ALY_JsonPayload ▶ 134	Abrufen der Länge der JSON-Nutzlast.
GetJsonString	Vererbt von T_ALY_JsonPayload ▶ 134	Abrufen der JSON-Nutzlast als String.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.25	PC oder CX (x64, x86, Arm®)	Tc3_AnalyticsStorageProvider

7.1.1.2.3 T_ALY_SPRecordData_Desc



Syntax

Definition:

```
FUNCTION_BLOCK T_ALY_SPRecordData_Desc EXTENDS T_ALY_JsonPayload
VAR_OUTPUT
    sRecordDataGuid    : GUID;
    nStartTimestamp    : LINT;
    eStatus             : E_RecordingState;
    nRecordID          : LINT;
    stRecord           : T_ALY_SPRecordData_Cmd;
END_VAR
```

Vererbungshierarchie

T_ALY_JsonPayload [[▶ 134](#)]

T_ALY_SPRecordData_Desc

Ausgänge

Name	Typ	Beschreibung
sRecordDataGuid	GUID	GUID „Key“ zum Identifizieren der Aufnahme
nStartTimestamp	LINT	Startzeit der Aufnahme.
eStatus	E_RecordingState [▶ 148]	Status der Aufnahme
nRecordID	LINT	Aufnahme ID
stRecord	T_ALY_SPRecordData_Cmd [▶ 118]	Zugehörige Aufnahme Kommando

Methoden

Name	Definitionsart	Beschreibung
Reset		Zurücksetzen aller Werte im Nutzlast-FB.
Init_JsonValue	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-Objekt.
Init_String	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-String.
GetJsonLength	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der Länge der JSON-Nutzlast.
GetJsonString	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der JSON-Nutzlast als String.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.25	PC oder CX (x64, x86, Arm®)	Tc3_AnalyticsStorageProvider

7.1.1.2.4 T_ALY_SPRecording_Desc

T_ALY_SPRecording_Desc

ARRAY [0..99] OF T_ALY_SPRecordData_Desc arrRecordings

Syntax

Definition:

```
FUNCTION_BLOCK T_ALY_SPCRecording_Desc EXTENDS T_ALY_JsonPayload
VAR_OUTPUT
    aRecordings : ARRAY [0..99] OF T_ALY_SPCRecordData_Desc;
END_VAR
```

Vererbungshierarchie

T_ALY_JsonPayload [[▶ 134](#)]

T_ALY_SPCRecording_Desc

 **Ausgänge**

Name	Typ	Beschreibung
aRecordings	ARRAY [0..99] OF T_ALY_SPCRecordData_Desc [▶ 125]	Liste aller aktuell laufenden Aufnahmen

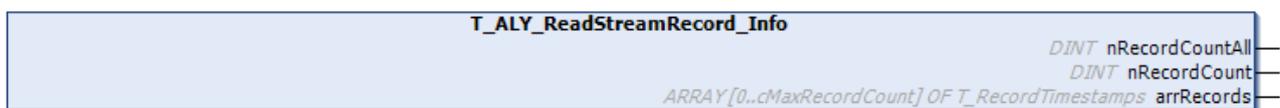
 **Methoden**

Name	Definitionsart	Beschreibung
Reset		Zurücksetzen aller Werte im Nutzlast-FB.
Init_JsonValue	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-Objekt.
Init_String	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-String.
GetJsonLength	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der Länge der JSON-Nutzlast.
GetJsonString	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der JSON-Nutzlast als String.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.25	PC oder CX (x64, x86, Arm®)	Tc3_AnalyticsStorageProvider

7.1.1.3 Info

7.1.1.3.1 T_ALY_ReadStreamRecord_Info



Syntax

Definition:

```
FUNCTION_BLOCK T_ALY_HistoricalStream_Desc EXTENDS T_ALY_JsonPayload
VAR_OUTPUT
    nRecordCountAll: UDINT;
    nRecordCount : UDINT;
    arrRecords : ARRAY [0..cMaxRecordCount] OF T_RecordTimestamps;
END_VAR
```

Vererbungshierarchie

T_ALY_JsonPayload [[▶ 134](#)]

T_ALY_ReadStreamRecord_Info

Ausgänge

Name	Typ	Beschreibung
nRecordCountAll	UDINT	Anzahl aller vorhanden Aufzeichnungen
nRecordCount	UDINT	Anzahl ausgelesener Aufzeichnungen
arrRecords	ARRAY [0..cMaxRecordCount] OF T_RecordTimestamps [▶ 133]	Zeitstempel der ausgelesenen Aufzeichnungen

Methoden

Name	Definitionsort	Beschreibung
Reset		Zurücksetzen aller Werte im Nutzlast-FB.
Init_JsonValue	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-Objekt.
Init_String	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-String.
GetJsonLength	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der Länge der JSON-Nutzlast.
GetJsonString	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der JSON-Nutzlast als String.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.25	PC oder CX (x64, x86, Arm®)	Tc3_AnalyticsStorageProvider

7.1.1.4 SubTypes

7.1.1.4.1 T_ALY_SPDataStorageInfo

T_ALY_SPDataStorageInfo
<i>T_ALY_SPStorageInfo</i> stStorage
<i>E_StorageState</i> eStatus
<i>STRING(255)</i> sStatusMessage

Syntax

Definition:

```
FUNCTION_BLOCK T_ALY_SPDataStorageInfo EXTENDS T_ALY_JsonPayload
VAR_OUTPUT
    stStorage      : T_ALY_SPStorageInfo;
    eStatus        : E_StorageState;
    sStatusMessage : STRING(255);
END_VAR
```

Vererbungshierarchie

T_ALY_JsonPayload [[▶ 134](#)]

T_ALY_SPDataStorageInfo

Ausgänge

Name	Typ	Beschreibung
stStorage	T_ALY_SPStorageInfo [▶ 130]	Detaillierte Storage Information
eStatus	E_StorageState [▶ 150]	Status des Storage
sStatusMessage	STRING(255)	Storage Status Meldung

Methoden

Name	Definitionsart	Beschreibung
Reset		Zurücksetzen aller Werte im Nutzlast-FB.
Init_JsonValue	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-Objekt.
Init_String	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-String.
GetJsonLength	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der Länge der JSON-Nutzlast.
GetJsonString	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der JSON-Nutzlast als String.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.25	PC oder CX (x64, x86, Arm®)	Tc3_AnalyticsStorageProvider

7.1.1.4.2 T_ALY_SPInstanceInfo

T_ALY_SPInstanceInfo

GUID sProviderGuid

STRING sServiceType

STRING(255) sDataStoreType

STRING(255) sComment

GUID sDefaultStorageGuid

ARRAY [0..49] OF T_ALY_SPDataStorageInfo arrDataStorages

Syntax

Definition:

```
FUNCTION_BLOCK T_ALY_SPInstanceInfo EXTENDS T_ALY_JsonPayload
VAR_OUTPUT
    sProviderGuid      : GUID;
    sServiceType       : STRING;
    sDataStoreType     : STRING(255);
    sComment           : STRING(255);
    sDefaultStorageGuid : GUID;
    arrDataStorages   : ARRAY [0..49] OF T_ALY_SPDataStorageInfo;
END_VAR
```

Vererbungshierarchie

T_ALY_JsonPayload [▶ 134]

T_ALY_SPInstanceInfo

Ausgänge

Name	Typ	Beschreibung
sProviderGuid	GUID	Individuelle GUID einer Storage Provider Instanz
sServiceType	STRING	Service Typ
sDataStoreType	STRING(255)	Storage Typ
sComment	STRING(255)	Kommentar zur Storage Provider Instanz
sDefaultStorageGuid	GUID	Storage GUID des Standard Storage
arrDataStorages	ARRAY [0..49] OF T_ALY_SPDataStorageInfo [▶ 128]	Liste der konfigurierten Storages

Methoden

Name	Definitionort	Beschreibung
Reset		Zurücksetzen aller Werte im Nutzlast-FB.
Init_JsonValue	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-Objekt.
Init_String	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-String.
GetJsonLength	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der Länge der JSON-Nutzlast.
GetJsonString	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der JSON-Nutzlast als String.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.25	PC oder CX (x64, x86, Arm®)	Tc3_AnalyticsStorageProvider

7.1.1.4.3 T_ALY_SPStorageInfo



Syntax

Definition:

```

FUNCTION_BLOCK T_ALY_SPStorageInfo EXTENDS T_ALY_JsonPayload
VAR_OUTPUT
  sStorageGuid      : GUID;
  sStorageName      : STRING;
  eDataStorageType  : E_DataStorageType;
  sComment          : STRING(255);
END_VAR
  
```

Vererbungshierarchie

[T_ALY_JsonPayload](#) [▶ 134]

T_ALY_SPStorageInfo

Ausgänge

Name	Typ	Beschreibung
sStorageGuid	GUID	Individuelle GUID eines Storage
sStorageName	STRING	Name des Storage
eDataStorageType	E_DataStorageType [▶ 147]	Storage Typ
sComment	STRING(255)	Kommentar zum Storage

Methoden

Name	Definitionsort	Beschreibung
Reset		Zurücksetzen aller Werte im Nutzlast-FB.
Init_JsonValue	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-Objekt.
Init_String	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-String.
GetJsonLength	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der Länge der JSON-Nutzlast.
GetJsonString	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der JSON-Nutzlast als String.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.25	PC oder CX (x64, x86, Arm®)	Tc3_AnalyticsStorageProvider

7.1.1.4.4 T_ALY_SPSubBrokerInfo



Syntax

Definition:

```

FUNCTION_BLOCK T_ALY_SPSubBrokerInfo EXTENDS T_ALY_JsonPayload
VAR_OUTPUT
    sAlias          : STRING(255);
    sBrokerGuid    : GUID;
    sBrokerHost    : STRING(255);
    nBrokerPort    : DINT;
    bSecure        : BOOL;
END_VAR
    
```

Vererbungshierarchie

T_ALY_JsonPayload [▶ 134]

T_ALY_SPSubBrokerInfo

Ausgänge

Name	Typ	Beschreibung
sAlias	STRING(255)	Alias Name der Broker Konfiguration
sBrokerGuid	GUID	Individuelle GUID der Broker Konfiguraiton
sBrokerHost	STRING(255)	Broker Host Name
nBrokerPort	DINT	Broker Port
bSecure	BOOL	TRUE wenn Kommunikation über Zertifikate hergestellt wird.

Methoden

Name	Definitionsort	Beschreibung
Reset		Zurücksetzen aller Werte im Nutzlast-FB.
Init_JsonValue	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-Objekt.
Init_String	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-String.
GetJsonLength	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der Länge der JSON-Nutzlast.
GetJsonString	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der JSON-Nutzlast als String.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.25	PC oder CX (x64, x86, Arm®)	Tc3_AnalyticsStorageProvider

7.1.1.4.5 T_ALY_Symbol

T_ALY_Symbol	
—sName	<i>T_MaxString</i>
—sBaseType	<i>T_MaxString</i>
—nBitOffset	<i>UDINT</i>
—nBitSize	<i>UDINT</i>

Syntax

Definition:

```
FUNCTION_BLOCK T_ALY_Symbol EXTENDS T_ALY_JsonPayload
VAR_INPUT
    sName : T_MaxString;
    sBaseType : T_MaxString;
    nBitOffset : UDINT;
    nBitSize : UDINT;
END_VAR
```

Vererbungshierarchie

[T_ALY_JsonPayload](#) [[▶ 134](#)]

T_ALY_Symbol

 Eingänge

Name	Typ	Beschreibung
sName	T_MaxString	Name des Symbols
sBaseType	T_MaxString	Datentyp des Symbols
nBitOffset	UDINT	Bit-Offset des Symbols
nBitSize	UDINT	Bitgröße des Symbols

 Methoden

Name	Definitionsort	Beschreibung
Reset		Zurücksetzen aller Werte im Nutzlast-FB.
Init_JsonValue	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-Objekt.
Init_String	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-String.
GetJsonLength	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der Länge der JSON-Nutzlast.
GetJsonString	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der JSON-Nutzlast als String.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.25	PC oder CX (x64, x86, Arm®)	Tc3_AnalyticsStorageProvider

7.1.1.4.6 T_RecordTimestamps



Syntax

Definition:

```

FUNCTION_BLOCK T_RecordTimestamps EXTENDS T_ALY_JsonPayload
VAR_OUTPUT
    nRecordID : DINT;
    sAlias    : STRING(255);
    nStartTimestamp : LINT;
    nEndTimestamp : LINT;
END_VAR
    
```

Vererbungshierarchie

[T_ALY_JsonPayload](#) [\[▶ 134\]](#)

T_RecordTimestamps

Ausgänge

Name	Typ	Beschreibung
nRecordID	DINT	Nummer der Aufzeichnung
sAlias	STRING(255)	Alias-Name der Aufzeichnung
nStartTimestamp	LINT	Start-Zeitstempel der Aufzeichnung
nEndTimestamp	LINT	End-Zeitstempel der Aufzeichnung

Methoden

Name	Definitionsart	Beschreibung
Reset		Zurücksetzen aller Werte im Nutzlast-FB.
Init_JsonValue	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-Objekt.
Init_String	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-String.
GetJsonLength	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der Länge der JSON-Nutzlast.
GetJsonString	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der JSON-Nutzlast als String.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.25	PC oder CX (x64, x86, Arm®)	Tc3_AnalyticsStorageProvider

7.1.1.5 Base Types

7.1.1.5.1 T_ALY_JsonPayload

T_ALY_JsonPayload

Syntax

Definition:

```
FUNCTION_BLOCK INTERNAL T_ALY_JsonPayload
```

Methoden

Name	Definitionsart	Beschreibung
Reset		Zurücksetzen aller Werte im Nutzlast-FB.
Init_JsonValue	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-Objekt.
Init_String	Vererbt von T_ALY_JsonPayload [▶ 134]	Initialisierung des FB mit JSON-String.
GetJsonLength	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der Länge der JSON-Nutzlast.
GetJsonString	Vererbt von T_ALY_JsonPayload [▶ 134]	Abrufen der JSON-Nutzlast als String.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.25	PC oder CX (x64, x86, Arm®)	Tc3_AnalyticsStorageProvider

7.1.2 FB_ALY_StorageProvider



Der FB_ALY_StorageProvider ist ein Client-FB für die Kommunikation mit einer Storage Provider-Instanz. Der FB bietet Methoden, um historische Daten auszulösen oder Aufzeichnungen zu starten/stoppen.

Syntax

Definition:

```
FUNCTION_BLOCK FB_ALY_StorageProvider
VAR_INPUT
    stConfig : ST_ALY_SP_Config;
END_VAR
VAR_OUTPUT
    bBusy : BOOL;
    bError : BOOL;
    ipResultMessage : I_TcMessage;
    eConnectionState : ETcIotMqttClientState;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
stConfig	ST_ALY_SP_Config [▶ 145]	Struktur für die Konfiguration des FB.

Ausgänge

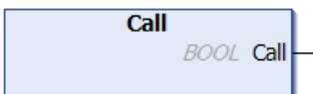
Name	Typ	Beschreibung
bBusy	BOOL	TRUE, sobald eine Methode des Funktionsbausteins aktiv ist.
bError	BOOL	Wird TRUE, sobald eine Fehlersituation auftritt.
ipResultMessage	I_TcMessage	Nachrichtenschnittstelle des TwinCAT 3 EventLoggers, die nähere Informationen über den Rückgabewert bietet.
eConnectionState	ETcIotMqttClientState	Gibt den Status der Verbindung zwischen Client und Broker als Aufzählung ETcIotMqttClientState an.

Methoden

Name	Definitionsort	Beschreibung
Call [▶ 136]	Local	Methode für Hintergrundkommunikation mit dem TwinCAT-Treiber. Die Methode muss zyklisch aufgerufen werden.
Cancel [▶ 137]	Local	Methode für den Abbruch von Aktivitäten des TwinCAT Analytics Storage Providers.
GetHistoricalData [▶ 137]	Local	Methode zum Anfordern historischer Daten.
GetInstanceInfo [▶ 137]	Local	Methode zum Empfangen der Instanz Informationen des Storage Providers.
GetRecordingInfoByAlias [▶ 138]	Lokal	Methode zum Empfangen von Recording Informationen.
GetRecordingInfoByKey [▶ 138]	Lokal	Methode zum Empfangen von Recording Informationen.
ReadHistoricalStreams [▶ 139]	Local	Methode zum Auslesen aller historischer Streams.
ReadStreamRecords [▶ 140]	Local	Methode zum Auslesen aller Aufnahmen eines historischen Streams.
ReadSubBroker [▶ 140]	Local	Methode zum Auslesen aller deklarierten Messagebroker.
ResetCommunication [▶ 141]	Local	Methode zum Zurücksetzen der MQTT Verbindung zum Broker.
RestartPipelineRule [▶ 141]	Local	Startet eine Rule einer Pipeline neu.
SendCommand [▶ 142]	Local	Generische Methode zum Absenden verschiedener Kommandos.
SetHistoricalDataState [▶ 142]	Local	Methode zum Setzen verschiedener Parameter eines historischen Streams.
StartPipeline [▶ 143]	Local	Startet die Aufzeichnung eines Live-MQTT-Binärstroms.
StartRecord [▶ 143]	Local	Startet die Aufzeichnung eines Live-MQTT-Binärstroms.
StartRecordEx [▶ 143]	Local	Startet die Aufzeichnung eines Live-MQTT-Binärstroms.
StopPipeline [▶ 144]	Local	Stoppt die ausgewählte Aufzeichnung.
StopRecord [▶ 144]	Local	Stoppt die ausgewählte Aufzeichnung.
StorageControlling [▶ 145]	Local	Methode zum Steuern der deklarierten Storages.

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.25	PC oder CX (x64, x86, Arm®)	Tc3_AnalyticsStorageProvider

7.1.2.1 Call



Syntax

```
METHOD Call : BOOL
```

Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

7.1.2.2 Cancel



Syntax

```
METHOD Cancel : BOOL
VAR_INPUT
    stCmd : REFERENCE TO T_ALY_SPCancel_Cmd;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
stCmd	REFERENCE TO T_ALY_SPCancel_Cmd [▶ 114]	JSON-Befehl zum Abbruch der Operationen des TwinCAT Analytics Storage Providers.

Rückgabewert

Name	Typ	Beschreibung
Cancel	BOOL	Ist TRUE, wenn abgeschlossen

7.1.2.3 GetHistoricalData



Syntax

```
METHOD GetHistoricalData : BOOL
VAR_INPUT
    stCmd : REFERENCE TO T_ALY_SPHistorical_Cmd;
END_VAR
```

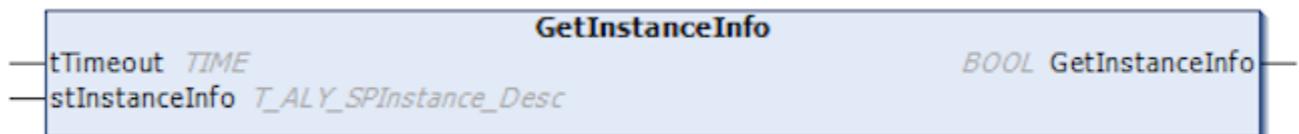
Eingänge

Name	Typ	Beschreibung
stCmd	REFERENCE TO T_ALY_SPGetHistorical_Cmd [▶ 115]	JSON-Befehl zum Abrufen historischer Daten von TwinCAT Analytics Storage Provider.

Rückgabewert

Name	Typ	Beschreibung
GetHistoricalData	BOOL	Ist TRUE, wenn abgeschlossen

7.1.2.4 GetInstanceInfo



Syntax

```

METHOD GetInstanceInfo : BOOL
VAR_INPUT
    tTimeout : TIME;
    stInstanceInfo : T_ALY_SPIInstance_Desc;
END_VAR

```

Eingänge

Name	Typ	Beschreibung
tTimeout	TIME	Dauer bis der Vorgang abgebrochen wird.
stInstanceInfo	T_ALY_SPIInstance_Desc [▶ 124]	JSON-Beschreibung der Storage Provider Instance.

Rückgabewert

Name	Typ	Beschreibung
GetInstanceInfo	BOOL	Ist TRUE, wenn abgeschlossen

7.1.2.5 GetRecordingInfoByAlias**Syntax**

```

METHOD GetRecordingInfoByAlias : BOOL
VAR_INPUT
    sRecordingAlias : STRING(255);
    tTimeout : TIME;
    stRecordingInfo : T_ALY_SPRecordData_Desc;
END_VAR

```

Eingänge

Name	Typ	Beschreibung
sRecordingAlias	STRING(255)	Suchkriterium „Alias“
tTimeout	TIME	Dauer bis der Vorgang abgebrochen wird.
stRecordingInfo	T_ALY_SPRecordData_Desc [▶ 125]	JSON-Beschreibung der Aufnahme.

Rückgabewert

Name	Typ	Beschreibung
GetRecordingInfoByAlias	BOOL	Ist TRUE, wenn abgeschlossen

7.1.2.6 GetRecordingInfoByKey

Syntax

```
METHOD GetRecordingInfoByKey : BOOL
VAR_INPUT
    sRecordDataKey : GUID;
    tTimeout        : TIME;
    stRecordingInfo : T_ALY_SPRecordData_Desc;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
sRecordDataKey	GUID	Suchkriterium „RecordDataKey“
tTimeout	TIME	Dauer bis der Vorgang abgebrochen wird.
stRecordingInfo	T_ALY_SPRecordData_Desc [▶ 125]	JSON-Beschreibung der Aufnahme.

 **Rückgabewert**

Name	Typ	Beschreibung
GetRecordingInfoByKey	BOOL	Ist TRUE, wenn abgeschlossen

7.1.2.7 ReadHistoricalStreams



Syntax

```
METHOD GetHistoricalData : BOOL
VAR_INPUT
    tSearchDuration : TIME := TIME#5s0ms
    aHistoricalStreams : POINTER TO T_ALY_HistoricalStream_Desc;
    nStreamCount : INT
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
tSearchDuration	TIME	Zeitspanne in der auf Rückmeldung gewartet wird.
aHistoricalStreams	POINTER TO T_ALY_HistoricalStream_Desc [▶ 123]	Beschreibung der verschieden historischen Streams

 **Rückgabewert**

Name	Typ	Beschreibung
ReadHistoricalStreams	BOOL	Ist TRUE, wenn abgeschlossen
nStreamCount	INT	Anzahl der ausgelesen Streams

7.1.2.8 ReadStreamRecords

ReadStreamRecords		
stCmd	REFERENCE TO T_ALY_SPReadStreamRecords_Cmd	BOOL ReadStreamRecords
tSearchTimeout	TIME	DINT nRecordCount
aStreamRecords	POINTER TO T_RecordTimestamps	

Syntax

```
METHOD ReadStreamRecords : BOOL
VAR_INPUT
    stCmd : REFERENCE TO T_ALY_SPReadStreamRecords_Cmd;
tSearchTimeout : TIME := TIME#5s0ms;
aStreamRecords : POINTER TO T_RecordTimestamps;
nRecordCount : DINT;
END_VAR
```

📌 Eingänge

Name	Typ	Beschreibung
stCmd	REFERENCE TO T_ALY_SPReadStreamRecords_Cmd [▶ 117]	JSON-Befehl zum Abrufen von Aufnahmen eines historischen Streams vom TwinCAT Analytics Storage Provider.
tSearchTimeout	TIME	Wartezeit auf die Antwort.
aStreamRecords	POINTER TO T_RecordTimestamps [▶ 133]	Ausgelesene Aufnahmen

📌 Rückgabewert

Name	Typ	Beschreibung
ReadStreamRecords	BOOL	Ist TRUE, wenn abgeschlossen
nRecordCount	DINT	Anzahl der ausgelesenen Aufnahmen

7.1.2.9 ReadSubBroker

ReadSubBroker		
tSearchDuration	TIME	BOOL ReadSubBroker
aSubBrokerInfos	POINTER TO T_ALY_SPSubBrokerInfo	INT nBrokerCount

Syntax

```
METHOD ReadSubBroker : BOOL
VAR_INPUT
    tSearchDuration : TIME;
    aSubBrokerInfos : POINTER TO T_ALY_SPSubBrokerInfo;
    nBrokerCount : INT;
END_VAR
```

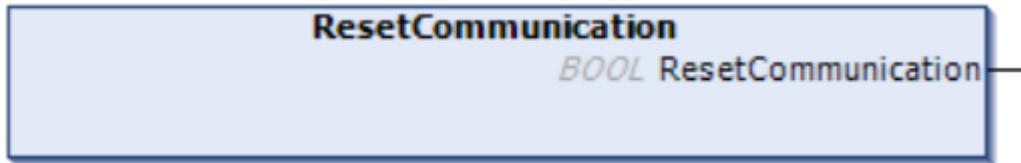
📌 Eingänge

Name	Typ	Beschreibung
tSearchDuration	TIME	Dauer bis die Suche beendet wird
aSubBrokerInfo	POINTER TO T_ALY_SPSubBrokerInfo [▶ 131]	Adresse zu einem Array in dem die gefundenen Broker Informationen gespeichert werden.
nBrokerCount	INT	Anzahl der gefundenen Broker

 Rückgabewert

Name	Typ	Beschreibung
ReadSubBroker	BOOL	Ist TRUE, wenn abgeschlossen

7.1.2.10 ResetCommunication



Syntax

```
METHOD ResetCommunication : BOOL
VAR_INPUT
END_VAR
```

 Rückgabewert

Name	Typ	Beschreibung
ResetCommunication	BOOL	Ist TRUE, wenn abgeschlossen

7.1.2.11 RestartPipelineRule



Syntax

```
METHOD RestartPipelineRule : BOOL
VAR_INPUT
    stCmd : REFERENCE TO T_ALY_SPRuleEnginePipeline_Cmd;
    nRuleId : DINT;
END_VAR
```

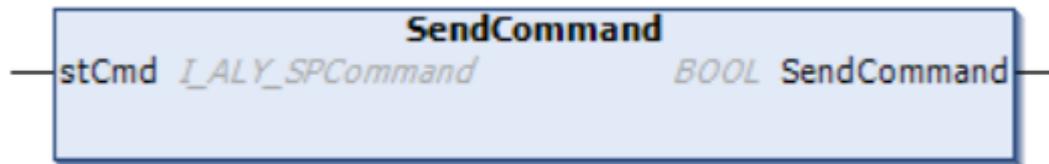
 Eingänge

Name	Typ	Beschreibung
stCmd	REFERENCE TO T_ALY_SPRuleEnginePipeline_Cmd ▶ 122	JSON-Befehl zum Starten der Aufzeichnung eines Live-Streams.
nRuleId	DINT	ID der neu zu startenden Rule.

 Rückgabewert

Name	Typ	Beschreibung
RestartPipelineRule	BOOL	Ist TRUE, wenn abgeschlossen.

7.1.2.12 SendCommand



Syntax

```
METHOD SendCommand : BOOL
VAR_INPUT
    stCmd : I_ALY_SPCCommand;
END_VAR
```

📌 Eingänge

Name	Typ	Beschreibung
stCmd	I_ALY_SPCCommand	JSON-Befehl zum Interagieren mit dem TwinCAT Analytics Storage Provider.

📌 Rückgabewert

Name	Typ	Beschreibung
SendCommand	BOOL	Ist TRUE, wenn abgeschlossen

7.1.2.13 SetHistoricalDataState



Syntax

```
METHOD SetHistoricalDataState : BOOL
VAR_INPUT
    stCmd : REFERENCE TO T_ALY_SPGetSetHistoricalDataState_Cmd;
END_VAR
```

📌 Eingänge

Name	Typ	Beschreibung
stCmd	REFERENCE TO T_ALY_SPGetSetHistoricalDataState_Cmd [▶ 120]	JSON-Befehl zum Setzen von Parametern eines gestarteten historischem Stream vom TwinCAT Analytics Storage Provider.

📌 Rückgabewert

Name	Typ	Beschreibung
SetHistoricalDataState	BOOL	Ist TRUE, wenn abgeschlossen

7.1.2.14 StartPipeline



Syntax

```
METHOD StartPipeline : BOOL
VAR_INPUT
    stCmd : REFERENCE TO T_ALY_SPRuleEnginePipeline_Cmd;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
stCmd	REFERENCE TO T_ALY_SPRuleEnginePipeline_Cmd [▶ 122]	JSON-Befehl zum Starten der Aufzeichnung eines Live-Streams.

Rückgabewert

Name	Typ	Beschreibung
StartPipeline	BOOL	Ist TRUE, wenn abgeschlossen.

7.1.2.15 StartRecord



Syntax

```
METHOD StartRecord : BOOL
VAR_INPUT
    stCmd : REFERENCE TO T_ALY_SPRecordData_Cmd;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
stCmd	REFERENCE TO T_ALY_SPRecordData_Cmd [▶ 118]	JSON-Befehl zum Starten der Aufzeichnung eines Live-Streams.

Rückgabewert

Name	Typ	Beschreibung
StartRecord	BOOL	Ist TRUE, wenn abgeschlossen

7.1.2.16 StartRecordEx



Im Gegensatz zur Methode [StartRecord](#) [▶ 143] kann hier ein `RecordDataKey` mitgegeben werden. Mit diesem Schlüssel kann die gestartete Aufnahme leichter wieder gefunden werden, um den Status der Aufnahme zu überprüfen. Hierfür kann mit der Methode [GetRecordingInfoByKey](#) [▶ 138] die Aufnahme Information abgerufen werden.

Syntax

```
METHOD StartRecordEx : BOOL
VAR_INPUT
    stCmd : REFERENCE TO T_ALY_SPRecordData_Cmd;
    sRecordDataKey : GUID;
END_VAR
```

🔗 Eingänge

Name	Typ	Beschreibung
stCmd	REFERENCE TO T_ALY_SPRecordData_Cmd [▶ 118]	JSON-Befehl zum Starten der Aufzeichnung eines Live-Streams.
sRecordDataKey	GUID	Guid „Key“ zum Identifizieren der gestarteten Aufnahme.

🔗 Rückgabewert

Name	Typ	Beschreibung
StartRecordEx	BOOL	Ist TRUE, wenn abgeschlossen

7.1.2.17 StopPipeline



Syntax

```
METHOD StopPipeline : BOOL
VAR_INPUT
    stCmd : REFERENCE TO T_ALY_SPRuleEnginePipeline_Cmd;
END_VAR
```

🔗 Eingänge

Name	Typ	Beschreibung
stCmd	REFERENCE TO T_ALY_SPRuleEnginePipeline_Cmd [▶ 122]	JSON-Befehl zum Stoppen der Aufzeichnung eines Live-Streams.

🔗 Rückgabewert

Name	Typ	Beschreibung
StopPipeline	BOOL	Ist TRUE, wenn abgeschlossen.

7.1.2.18 StopRecord



Syntax

```
METHOD StopRecord : BOOL
VAR_INPUT
    stCmd : REFERENCE TO T_ALY_SPRecordData_Cmd;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
stCmd	REFERENCE TO T_ALY_SPRecordData_Cmd [▶_118]	JSON-Befehl zum Stoppen der Aufzeichnung eines Live-Streams.

 **Rückgabewert**

Name	Typ	Beschreibung
StopRecord	BOOL	Ist TRUE, wenn abgeschlossen.

7.1.2.19 StorageControlling



Syntax

```
METHOD StorageControlling : BOOL
VAR_INPUT
    stCmd : REFERENCE TO T_ALY_SPStorageCtrl_Cmd;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
stCmd	REFERENCE TO T_ALY_SPStorageCtrl_Cmd [▶_122]	JSON-Befehl zum Steuern der Storages.

 **Rückgabewert**

Name	Typ	Beschreibung
StorageControlling	BOOL	Ist TRUE, wenn abgeschlossen

7.2 Datentypen

7.2.1 ST_ALY_SP_Config

Syntax

Definition:

```
TYPE ST_Msg :
STRUCT
    sMainTopic      : T_MaxString;
    sProviderGuid   : GUID;
    stConnSettings  : ST_ConnectionSettings
END_STRUCT
END_TYPE
```

Parameter

Name	Typ	Beschreibung
sMainTopic	T_MaxString	Das Main-Topic, wenn sich der TwinCAT Analytics Storage Provider auf dem Message-Broker befindet.
sProviderGuid	GUID	Die individuelle GUID der TwinCAT Analytics Storage Provider-Instanz.
stConnSettings	ST_ConnectionSettings [► 146]	MQTT-Verbindungseinstellungen für die Verbindung mit dem Message-Broker.

7.2.2 ST_ConnectionSettings**Syntax**

Definition:

```

TYPE ST_ConnectionSettings :
STRUCT
  sHostName   : T_MaxString;
  nHostPort   : UINT := 1883;
  sUserId     : T_MaxString;
  sPassword   : T_MaxString;
  bWithCertificate : BOOL := BOOL;
  sCA         : T_MaxString;
  sCert       : T_MaxString;
  sKey        : T_MaxString;
sKeyPwd : T_MaxString;
END_STRUCT
END_TYPE

```

Parameter

Name	Typ	Beschreibung
sHostName	T_MaxString	sHostName kann als Name oder als IP-Adresse festgelegt werden. Wenn keine Angaben gemacht werden, wird der lokale Host verwendet.
nHostPort	UINT	Hier kann der Hostport festgelegt werden: Der Standardwert ist 1883.
sUserId	T_MaxString	Optional kann ein Benutzername angegeben werden.
sPassword	T_MaxString	Hier kann ein Passwort für den Benutzernamen eingegeben werden.
bWithCertificate	BOOL	Falls TRUE, werden die Zertifikate für die Kommunikation verwendet.
sCA	T_MaxString	Zertifikat der Zertifizierungsstelle (CA).
sCert	T_MaxString	Client-Zertifikat für die Authentifizierung beim Broker.
sKey	T_MaxString	Privater Schlüssel des Clients.
sKeyPwd	T_MaxString	Passwort des privaten Schlüssels, falls zutreffend.

7.2.3 E_CancelType**Syntax**

Definition:

```

TYPE E_CancelType :
(
  HistoricalData := 0,

```

```
AllRecordData
) INT;
END_TYPE
```

Parameter

Name	Beschreibung
HistoricalData	Bricht den ausgewählten historischen Datenstrom ab.
AllRecordData	Bricht alle laufenden Aufzeichnungen ab.

7.2.4 E_ControlMode

Syntax

Definition:

```
TYPE E_ControlMode :
(
    Start := 0,
    Stop
    DeleteSettings
) INT;
END_TYPE
```

Parameter

Name	Beschreibung
Start	Aufstarten des „Storage“ wird angetriggert.
Stop	Herunterfahren des „Storage“ wird angetriggert.
DeleteSettings	„Storage“ Konfiguration soll gelöscht werden (funktioniert nur wenn der „Storage“ nicht online ist).

7.2.5 E_DataStorageType

Syntax

Definition:

```
TYPE E_DataStorageType :
(
    Empty := 0,
    AnalyticsFile
    AzureBlob
    MsSQL_Binary
    InfluxDB
    MsSQL_Plain
    CSVFile
) INT;
END_TYPE
```

Parameter

Name	Beschreibung
Empty	Unbekannter „Storage“ Typ
AnalyticsFile	Analytics File (TwinCAT Analytics eigenes Datenformat)
AzureBlob	Microsoft Azure Blob
MsSQL_Binary	Microsoft SQL Server (Daten in Binär)
InfluxDB	Influx 2.x Datenbank
MsSQL_Plain	Microsoft SQL Server (Daten in Klartext)
CSVFile	CSV-Datei

7.2.6 E_RawDataFormat

Syntax

Definition:

```
TYPE E_RawDataFormat :
(
    Bin := 0,
    Json
) INT;
END_TYPE
```

Parameter

Name	Beschreibung
Bin	Analytics Binary Stream Format.
Json	TwinCAT Json-Format (wird derzeit nicht unterstützt).

7.2.7 E_RecordingState

Syntax

Definition:

```
TYPE E_RecordingState :
(
    Not_Initialized := 0,
    Initializing
    RecordingCanceled
    Running
    Running_QueueHyst
    Stopping_ReceivingDataStopped
    RecordingDone
    WaitingForData
    Error
) INT;
END_TYPE
```

Parameter

Name	Beschreibung
Not_Initialized	Aufnahme angetriggert. Wartet auf Eingangsdatenbeschreibung zum Initialisieren.
Initialized	Aufnahme erfolgreich initialisiert.
RecordingCanceled	Aufnahme abgebrochen.
Running	Aufnahme läuft. Eingangsdaten werden gespeichert.
Running_QueueHyst	Aufnahme läuft. Eingangsdaten können nicht schnell genug gespeichert werden. Datenverlust!
Stopping_ReceivingDataStopped	Aufnahme wird beendet.
RecordingDone	Aufnahme beendet.
WaitingForData	Aufnahme läuft. Es kommen keine Daten vom Analytics Logger an.
Error	Es ist ein Fehler während der Aufnahme aufgetreten.

7.2.8 E_RecordMode

Syntax

Definition:

```
TYPE E_RecordMode :
(
    Start := 0,
    Stop
) INT;
END_TYPE
```

Parameter

Name	Beschreibung
Start	Startet die konfigurierte Aufzeichnung.
Stop	Stoppt die Aufzeichnung.

7.2.9 E_ReloadType

Syntax

Definition:

```
TYPE E_ReloadType :
(
    All := 0,
    Specific
) INT;
END_TYPE
```

Parameter

Name	Beschreibung
All	Es werden alle Aufzeichnungen neu eingelesen.
Specific	Es wird nur eine spezifische Aufnahme neu eingelesen.

7.2.10 E_RingBufferMode

Syntax

Definition:

```
TYPE E_RingBufferMode:
(
    None := 0,
    TimeBased,
    DataBased
) INT;
END_TYPE
```

Parameter

Name	Beschreibung
None	Aufzeichnung ohne Ringpuffermodus.
TimeBased	Ringpuffer basiert auf einem bestimmten Zeitraum.
DataBased	Ringbuffer basiert auf einer bestimmten maximalen Datengröße.

7.2.11 E_SetGetHistoricalDataState

Syntax

Definition:

```
TYPE E_SetGetHistoricalDataState :
(
    Pause,
```

```

    Continue_,
Restart,
Stop,
Update
)INT;
END_TYPE

```

Parameter

Name	Beschreibung
Pause	Abspielen der Aufzeichnung wird pausiert.
Continue_	Abspielen der Aufzeichnung wird fortgesetzt.
Restart	Abspielen der Aufzeichnung wird neu gestartet.
Stop	Abspielen der Aufzeichnung wird beendet.
Update	Parameter zum Abspielen der Aufzeichnung werden aktualisiert.

7.2.12 E_StorageState

Syntax

Definition:

```

TYPE E_StorageState :
(
    unknown := 0,
    error
    starting
    online
    shuttingDownt
    offline
)INT;
END_TYPE

```

Parameter

Name	Beschreibung
Unknown	Status des „Storage“ ist unbekannt
Error	„Storage“ befindet sich im Fehlerfall. Es werden keine Anfragen mehr verarbeitet.
Starting	Der „Storage“ startet auf und verbindet sich.
Online	Der „Storage“ läuft und ist bereit für Anfragen.
ShuttingDown	Der „Storage“ wird heruntergefahren.
Offline	Der „Storage“ ist aus und nicht erreichbar.

7.2.13 E_SymbolMode

Syntax

Definition:

```

TYPE E_SymbolMode :
(
    All := 0,
    Subset
)INT;
END_TYPE

```

Parameter

Name	Beschreibung
All	Alle Symbole des Streams werden verwendet.
Subset	Nur eine Teilmenge der Symbole wird verwendet.

7.2.14 E_PipelineCmdType**Syntax**

Definition:

```

TYPE E_PipelineCmdType :
(
  RestartRule := 5,
  Start := 6,
  Stop := 7
) INT;
END_TYPE

```

Parameter

Name	Beschreibung
RestartRule	Startet eine bestimmte Rule einer RuleEnginePipeline neu.
Start	Startet eine RuleEnginePipeline
Stop	Stoppt eine RuleEnginePipeline

8 Beispiele

8.1 SPS-Client

Dieses SPS-Beispiel zeigt die Verwendung der TwinCAT Analytics Storage Provider Bibliothek. Der Beispiel-Code zeigt das Lesen und Schreiben. Damit das Beispiel in sich geschlossen funktioniert, wird sowohl die Verwendung des Analytics Loggers für das Schicken von Messdaten an einen MQTT Message Broker gezeigt wie auch das Einspielen der historischen Daten über den Analytics Stream Helper.

Grundlage ist ein entsprechend aufgesetzter nativer MQTT Message Broker und ein Analytics Storage Provider Service.

Das SPS-Beispiel zeigt folgende Ablaufschritte:

1. Analytics Logger: Stream von Variablen aus einer Globalen Variablen Liste an einen MQTT Message Broker.
2. Analytics Storage Provider: Starten und stoppen von Stores und Aufnahmen, sowie Lesen von Aufnahmen und Lesen der historischen Daten.
3. Analytics Stream Helper: Empfangen der historischen Daten vom Analytics Storage Provider und das Mappen der Daten in eine Globale Variablen Liste für die historischen Daten.

Analytics Storage Provider GUID Glossar

Bei der Verwendung des Storage Providers treten verschiedene GUIDs auf, die zur Identifikation von Services und Daten notwendig sind. Nachfolgend wird beschrieben, woher die GUIDs stammen, welchen Zweck sie erfüllen und wo sie ggf. eingesehen werden können.

Der Analytics Daten-Stream welcher vom Analytics Logger verschickt wird, wird im Grunde von drei Parametern beschrieben:

1. **Topic** [STRING]
Wohin werden die Daten geschickt?
2. **TwinCAT SystemID** [GUID]
Von welchem TwinCAT System werden die Daten verschickt?
3. **LayoutID / Symbol Info ID** [GUID]
Wie sehen die Daten aus? Die GUID ist ein Hash über die Symbolinformationen.



Um Aufzeichnungen am Storage Provider zu identifizieren, werden die oben genannten Parameter benötigt.

Auf einem MQTT Message Broker können mehrere Storage Provider verbunden sein. Um einen Storage Provider zu identifizieren, werden zwei Parameter benötigt.

1. **MainTopic** [STRING](Wo werden die Daten/Services bereitgestellt?)
2. **ProviderGuid** [GUID](Eindeutige Kennung des Services)

Um zu erkennen, wer eine Aufnahme am Storage Provider gestartet hat, gibt es die **RecorderGuid**. Diese GUID wird automatisch an jedem Storage Provider Manager oder Client generiert und an den StartRecord-Kommandos angehängt. In der SPS kann diese frei am StartRecord-Kommando vergeben werden.

Ab der Version 3.2.14 können am StartRecord-Kommando drei weitere GUIDs angegeben werden.

1. **Storage** [GUID]
Diese GUID gibt an, in welchen Storage die Daten abgespeichert werden sollen. Die GUID wird im Analytics Storage Provider Configurator automatisch generiert. Sie kann dort, oder im Analytics Storage Provider Manager ausgelesen werden. Wird keine GUID angegeben wird der Master Storage verwendet.
2. **SubBroker/DataBroker** [GUID]
Diese GUID gibt an, von welchem Data Broker der Analytics Stream entgegengenommen werden soll. Der Analytics Storage Provider bietet die Möglichkeit, von mehreren Messagebrokern aufzuzeichnen.

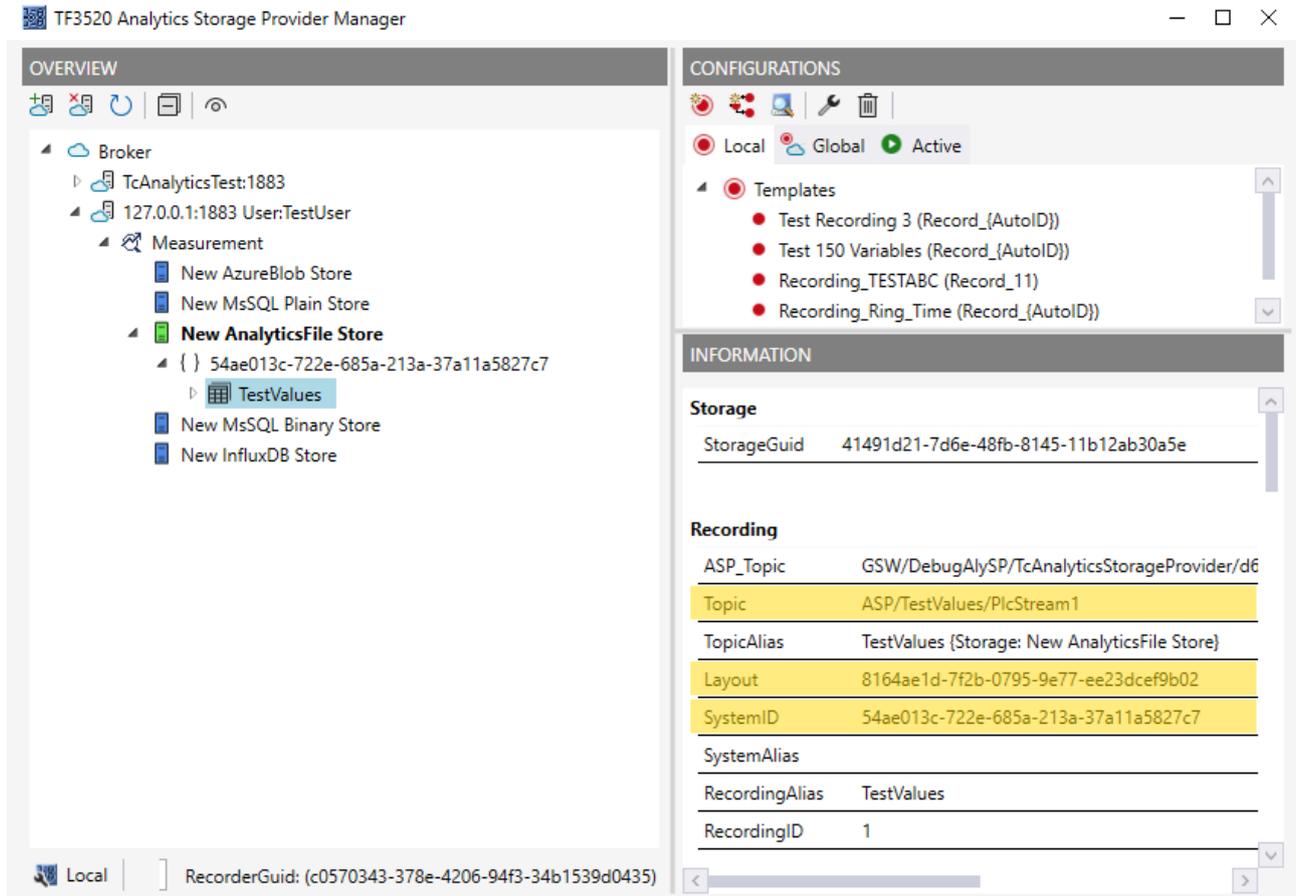
Die GUID wird im Analytics Storage Provider Configurator automatisch generiert. Sie kann dort oder im Analytics Storage Provider Manager ausgelesen werden. Wird keine GUID angegeben, wird der Master Data Broker verwendet.

3. **DataKey** [GUID]

Mithilfe des DataKeys können laufende Aufnahmen gefunden, ausgelesen und überwacht werden. Dieser DataKey kann in der SPS frei gewählt werden. Wird kein DataKey angegeben, wird automatisch vom Storage Provider ein DataKey generiert.

Nachfolgend einige Screenshots, die die oben beschriebenen Parameter und GUIDs enthalten.

Topic / TwinCAT System ID / Layout (System Info ID)



Storage Provider MainTopic / ProviderGuid / RecorderGuid

TF3520 Analytics Storage Provider Manager

Local
RecorderGuid: (c0570343-378e-4206-94f3-34b1539d0435)

OVERVIEW

- Broker
 - TcAnalyticsTest:1883
 - 127.0.0.1:1883 User:TestUser
 - Measurement
 - New AzureBlob Store
 - New MsSQL Plain Store
 - New AnalyticsFile Store**
 - New MsSQL Binary Store
 - New InfluxDB Store

CONFIGURATIONS

Local Global Active

Templates

- Test Recording 3 (Record_{AutoID})
- Test 150 Variables (Record_{AutoID})
- Recording_TESTABC (Record_11)
- Recording_Ring_Time (Record_{AutoID})

INFORMATION

Provider

Topic	GSW/Analytics SP/TcAnalyticsStorageProvider/d67d1
Version	1.0.0
Name	TwinCAT Analytics Storage Provider
Timestamp	08.10.2024 10:38:20

Provider Info

ProviderGuid	d67d1631-b890-4c5b-b5e0-1384549b05e2
DataStoreType	MultiStorageProvider {DefaultStorage: 4149}
ServiceType	Windows Service
Comment	Measurement
DefaultStorage_Guid	41491d21-7d6e-48fb-8145-11b12ab30a5e

Storage GUID

TF3520 Analytics Storage Provider Manager

OVERVIEW

- Broker
 - TcAnalyticsTest:1883
 - 127.0.0.1:1883 User:TestUser
 - Measurement
 - New AzureBlob Store
 - New MsSQL Plain Store
 - New AnalyticsFile Store**
 - New MsSQL Binary Store
 - New InfluxDB Store

CONFIGURATIONS

Local Global Active

Templates

- Test Recording 3 (Record_{AutoID})
- Test 150 Variables (Record_{AutoID})
- Recording_TESTABC (Record_11)
- Recording_Ring_Time (Record_{AutoID})

INFORMATION

Storage

StorageName	New AnalyticsFile Store
StorageType	AnalyticsFile
Comment	
StorageGuid	41491d21-7d6e-48fb-8145-11b12ab30a5e

Data Broker GUID

TF3520 Analytics Storage Provider Manager

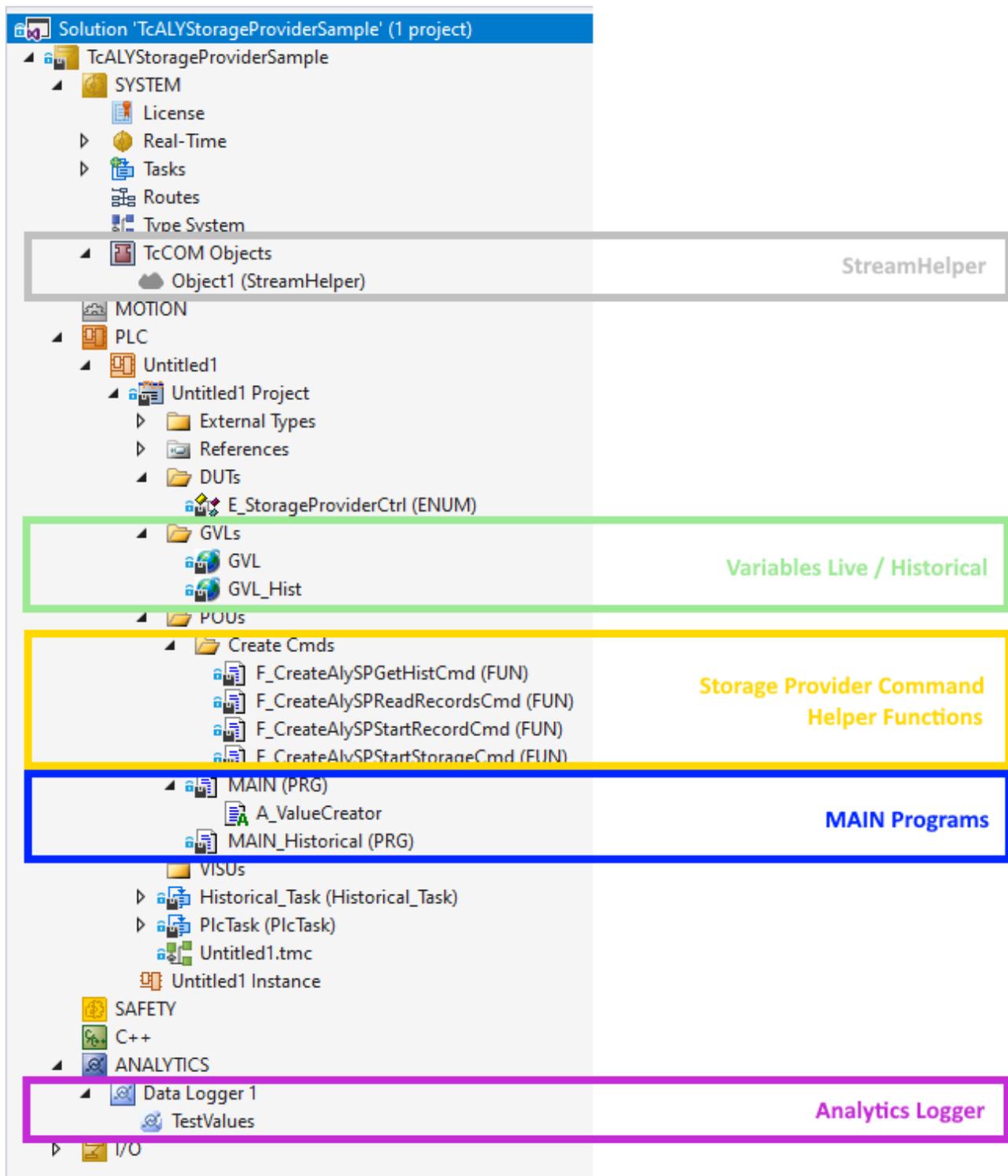
The screenshot shows the 'TF3520 Analytics Storage Provider Manager' interface. It is divided into three main sections: Overview, Configurations, and Information.

- OVERVIEW:** A tree view showing the hierarchy of brokers. Under '127.0.0.1:1883 User:TestUser', there is a 'Measurement' folder containing several stores: 'New AzureBlob Store', 'New MsSQL Plain Store', 'New AnalyticsFile Store' (highlighted in blue), 'New MsSQL Binary Store', and 'New InfluxDB Store'.
- CONFIGURATIONS:** Shows the status of the broker. 'Local' is selected, 'Global' is disabled, and 'Active' is enabled. Under 'Measurement', there is a 'Configurations' folder containing 'DataBroker', 'Nicolas_Broker', and 'LocalBroker' (highlighted in blue).
- INFORMATION:** A table titled 'Remote Broker' with the following data:

BrokerGuid	2d7b6394-500e-4dc2-8f81-2e800a12c7be
Alias	LocalBroker
Broker	127.0.0.1 : 1883
TopicList	

Beispiel-Code-Architektur

Alle relevanten Stellen der Konfiguration und des Programmcodes sind in dem nachfolgenden Bild markiert:



Stream Helper

Für den Empfang der historischen Daten, die von dem Analytics Storage Provider per MQTT verschickt werden.

Variable Live/Historisch

Die GVL ist für die Live-Daten und die GVL_Hist ist für die historischen Daten.

Storage Provider Command Helper Functions

Diese Helper-Funktionen erzeugen die Kommandos für die Kommunikation mit dem Storage Provider Service im JSON-Format.

MAIN Programm

Das Main-Programm ruft die Kommunikation zu dem Analytics Storage Provider auf. Das Main-Historical-Programm realisiert das Mapping der historischen Daten von dem Stream Helper in die GVL_Hist.

Analytics Logger

Sendet die Variablen der GVL an einen MQTT Message Broker.

Beispiel Starten

HINWEIS

Zu geringer Router Memory kann zu System Abstürzen führen

Erhöhen Sie den Router Memory in den Echtzeit Einstellungen auf 256 MB. Zusätzlich wird empfohlen, die maximale Stack Size der Global Task Konfiguration auf 512KB zu erhöhen.

Bevor das Beispiel gestartet werden kann, müssen Sie an drei unterschiedlichen Stellen den von Ihnen verwendeten MQTT Message Broker einstellen.

Analytics Stream Helper:

Object	Context	Parameter (Init)	Parameter (Online)	Interfaces		
		Name	Value	CS	Type	PTCID
-		MQTT				
		Host Name	172.17.62.145	<input type="checkbox"/>	STRING(80)	0x020201...
		TcpPort	1883	<input type="checkbox"/>	UINT	0x020201...
		Topic Prefix	_AlySPTest/ResultValues	<input type="checkbox"/>	STRING(255)	0x020300...
		Client ID		<input type="checkbox"/>	STRING(80)	0x020201...
		Username		<input type="checkbox"/>	STRING(80)	0x020201...
		Password		<input type="checkbox"/>	STRING(80)	0x020201...
		Max Handles	100	<input type="checkbox"/>	UDINT	0x020300...
		Fifo Exponent	8	<input type="checkbox"/>	UDINT	0x020300...
		Sender System Id		<input type="checkbox"/>	STRING(80)	0x020300...
-		Tls				
		CaFile		<input type="checkbox"/>	STRING(255)	0x020201...
		CertFile		<input type="checkbox"/>	STRING(255)	0x020201...
		KeyFile		<input type="checkbox"/>	STRING(255)	0x020201...
		KeyPw		<input type="checkbox"/>	STRING(255)	0x020201...
		CrlFile		<input type="checkbox"/>	STRING(255)	0x020201...
		PskId		<input type="checkbox"/>	STRING(255)	0x020201...
		PskPw		<input type="checkbox"/>	STRING(255)	0x020201...
		Insecure	FALSE	<input type="checkbox"/>	BOOL	0x020201...
		Version		<input type="checkbox"/>	STRING(255)	0x020201...

Analytics Logger:

Context	Parameter (Int)	Data Area	TLS	Time Source			
	Name	Value	CS	Type	PTCID		
	Data Format	ANALYTICS_FORMAT_MQTT_BINARY	<input type="checkbox"/>	ANALYTICS_FORMAT	0x02020114		
	Data Compression	ANALYTICS_COMPRESSION_RL	<input type="checkbox"/>	ANALYTICS_COMPRESS...	0x02030027		
	Max. Compression Compare Width	ANALYTICS_COMP_WIDTH_8	<input type="checkbox"/>	ANALYTICS_COMPRESS...	0x02030030		
	MQTT Host Name	172.17.62.145	<input type="checkbox"/>	STRING(80)	0x02020113		
	MQTT Tcp Port	1883	<input type="checkbox"/>	UINT	0x02020103		
	MQTT Main Topic	AlySPTest	<input type="checkbox"/>	STRING(255)	0x02030008		
	MQTT Client ID		<input type="checkbox"/>	STRING(80)	0x02020101		
	MQTT User Name		<input type="checkbox"/>	STRING(255)	0x02020106		
	MQTT Password		<input type="checkbox"/>	STRING(80)	0x02020107		

MAIN-Programm:

```

MAIN -> X
28 VAR
29   nState: INT;
30
31   fbALY_StorageProvider: FB_ALY_StorageProvider := (stConfig := (sMainTopic := 'GSW/Analytics SP',
32     sProviderGuid := STRING_TO_GUID('d67d1631-b890-4c5b-b5e0-1384549b05e2'),
33     stConnSettings := (sHostName := '172.17.64.145',
34       sUserID := 'TestUser', sPassword := '123',
35       sClientId := 'PLCClient'));
36
37   sLiveStreamTopic: T_MaxString := 'AlySPTest/TestValues';
38   sRecordingAlias: T_MaxString := 'PLC Recording';
39   sRecordName: T_MaxString := 'TestValueRec_(AutoID)';
40
41   stStorageCtrlCmd: T_ALY_SPStorageCtrl_Cmd;
42   stStartCmd: T_ALY_SPRecordData_Cmd;
43   stStopCmd: T_ALY_SPRecordData_Cmd;
44   stReadRecordsCmd: T_ALY_SPReadStreamRecords_Cmd;
45   stGetHistDataCmd: T_ALY_SPGetHistorical_Cmd;
46
47   sDataKey: GUID := STRING_TO_GUID('12345678-ABCD-1234-ABCD-123456789ABC');
48 END VAR

```

Anschließend müssen Sie das sMainTopic und die sProviderGUID für den FB_ALY_StorageProvider ändern. Zu finden ist diese wie oben in diesem Dokument beschrieben.

```

MAIN -> X
28 VAR
29   nState: INT;
30
31   fbALY_StorageProvider: FB_ALY_StorageProvider := (stConfig := (sMainTopic := 'GSW/Analytics SP',
32     sProviderGuid := STRING_TO_GUID('d67d1631-b890-4c5b-b5e0-1384549b05e2'),
33     stConnSettings := (sHostName := '172.17.64.145',
34       sUserID := 'TestUser', sPassword := '123',
35       sClientId := 'PLCClient'));
36
37   sLiveStreamTopic: T_MaxString := 'AlySPTest/TestValues';
38   sRecordingAlias: T_MaxString := 'PLC Recording';
39   sRecordName: T_MaxString := 'TestValueRec_(AutoID)';
40
41   stStorageCtrlCmd: T_ALY_SPStorageCtrl_Cmd;
42   stStartCmd: T_ALY_SPRecordData_Cmd;
43   stStopCmd: T_ALY_SPRecordData_Cmd;
44   stReadRecordsCmd: T_ALY_SPReadStreamRecords_Cmd;
45   stGetHistDataCmd: T_ALY_SPGetHistorical_Cmd;
46
47   sDataKey: GUID := STRING_TO_GUID('12345678-ABCD-1234-ABCD-123456789ABC');
48 END VAR

```

Für die Steuerung des Beispiels gehen Sie nun in das MAIN Programm. Mit dem Enum eCtrl können Sie die Aktion einstellen, die Sie durchführen möchten. Es stehen zur Auswahl:

- ReadASPDescription (Beinhaltet auch die Storage Beschreibung)
- ReadDataBroker
- StartStorage
- ShutdownStorage
- StartRecord
- StopRecord
- IsRecordingRunning
- ReadHistoricalStreams

- ReadRecords
- GetHistorical

Mit einer steigenden Flanke an der Variablen bExecute wird die im Enum gewählte Aktion ausgeführt. Sollten Sie mehrere Records gemacht haben, können Sie dies in dem Array aRecordInfo einsehen. Mit dem Index ist es dann möglich, die unterschiedlichen Records auszuwählen. Die Zeitspannen werden ebenfalls angezeigt, diese könnten Sie theoretisch auch noch innerhalb der Zeitspanne anpassen. Dafür müssten Sie die Logik des Beispiels in der Helfer-Funktion F_CreateAlySPGetHistCmd entsprechend anpassen.

Die oben im Dokument herausgesuchte Storage Provider Recorder GUID kann optional in der SPS in der Funktion F_CreateAlySPStartRecordCmd gesetzt werden. Theoretisch kann es eine beliebige GUID sein, sie dient nur der Identifikation des Recorders.

Download: https://infosys.beckhoff.com/content/1031/tf3500_tc3_analytics_logger/Resources/11270100747.zip

9 Anhang

9.1 Glossar

In der nachfolgenden Tabelle werden wesentliche Begriffe im Zusammenhang mit dem Storage Provider erläutert.

Name	Funktion	Eindeutige Kennung (automatisch generiert)	Beschreibende Parameter (durch Anwender konfigurierbar)
Storage Provider Configurator	Softwarewerkzeug zur Konfiguration des lokalen Storage Providers.		
Storage Provider Manager	Softwarewerkzeug zum Arbeiten mit dem Storage Provider. Hierüber lassen sich Datenaufnahmen starten, stoppen und verwalten. Der Storage Provider Manager kann sowohl für den lokalen als auch für den remote Storage Provider verwendet werden.		
Storage Provider CLI Client	Kommandozeilentool für Interaktionen mit dem Storage Provider.		
Storage Provider SPS-Bibliothek	SPS-Bibliothek für Interaktionen mit dem Storage Provider.		
Storage Provider	Softwareanwendung zum Historisieren von Daten wie z. B. vom Analytics Logger. Sowohl die Akquisition der zu speichernden Daten als auch die Bereitstellung von gespeicherten Daten erfolgt über MQTT. Der Storage Provider kann unter Windows sowie FreeBSD® betrieben werden.	ProviderGuid	ProviderAlias
Storage	Datensenke eines Storage Providers (z. B. MS SQL oder CSV).	StorageGuid	StorageAlias
Message Broker	MQTT-Messagebroker über den mit dem MQTT-Protokoll Daten übermittelt werden können.		
HostBroker	Zentraler Message Broker, auf welchem der Storage Provider Informationen bereitstellt und Kommandos entgegennimmt.	BrokerGuid	BrokerAlias
DataBroker	Zusätzlicher Message Broker, von welchem der Storage Provider aufzunehmende Daten beziehen kann.	BrokerGuid	BrokerAlias
Pipeline	Beschreibung eines Datenflusses. Bestandteile sind Datenquellen (Input Sources), Verarbeitungsschritte (Rules) und Datenablagen (Recordings).	PipelineGuid	PipelineAlias (kann vor dem Aufstarten der Pipeline gesetzt werden)
RuleEngine	Verarbeitungseinheit innerhalb des Storage Providers.		
Rule	Verarbeitungsregel innerhalb einer Pipeline, welche für die Filterung, Aggregation und Sampling von Daten dient.	RuleID	RuleAlias
Recording	Aufnahmekonfiguration für Daten. Diese umfasst, welche Daten in welchen Storage aufgezeichnet werden sollen.	RecordingID	RecordingAlias (kann vor dem Aufstarten der Pipeline gesetzt werden)
Record	Datenaufnahme, welche auf einer definierten Aufnahmekonfiguration (Recording) basiert.	RecordID, bei laufender Aufnahme zusätzlich RecordDataKey	RecordAlias (kann vor dem Aufstarten der Pipeline gesetzt werden)
Input Source	Datenquelle für eine Pipeline. Das kann ein MQTT-LiveStream oder ein HistoricalStream sein.		

Name	Funktion	Eindeutige Kennung (automatisch generiert)	Beschreibende Parameter (durch Anwender konfigurierbar)
LiveStream	Daten vom Analytics Logger, IoT Data Agent oder EK9160.		
HistoricalStream	MQTT-Datenstrom vom Storage Provider, welcher alle Infos zu einem Recording und den dazugehörigen Records umfasst. Pro Recording wird ein HistoricalStream erzeugt.	HistStreamID	HistStreamAlias - dieser entspricht dem RecordingAlias
Recorder	Kennzeichnung des Clients, der mit dem Provider kommuniziert. Sie wird beim Start der Pipeline mitgegeben, um nachvollziehen zu können, durch wen die Pipeline gestartet wurde.	RecorderGuid	RecorderAlias

9.2 FAQ – Häufig gestellte Fragen und Antworten

In diesem Abschnitt werden häufig gestellte Fragen beantwortet, um Ihnen die Arbeit mit dem TwinCAT Analytics Storage Provider (ASP) zu erleichtern. Falls Sie weitere Fragen haben, wenden Sie sich bitte an unser Support-Team support@beckhoff.com.

1. [Wie kann ich das Tabellenschema von MS SQL mit ASP verwalten? \[► 162\]](#)
2. [Kann ich den Storage Provider auf programmierbare Weise steuern? \[► 162\]](#)
3. [Ist es möglich, auch Ergebnisse von der Analytics Runtime zu speichern? \[► 162\]](#)
4. [Werden in TwinCAT Measurement Produkten Open source Softwarekomponenten verwendet? \[► 162\]](#)
5. [Von welchen Einflussfaktoren hängt der Datendurchsatz des Storage Providers ab? \[► 162\]](#)

Wie kann ich das Tabellenschema von MS SQL mit ASP verwalten?

Über das Tabellenschema müssen Sie sich keine Gedanken machen. Dies wird komplett vom Analytics Storage Provider übernommen. Sie müssen lediglich festlegen, auf welchem Datenbankserver die Daten gespeichert werden sollen. Wenn Sie Daten in Ihrer eigenen Tabellenstruktur sehen möchten, müssen Sie die Daten in eine TwinCAT Analytics Runtime streamen und vom TwinCAT Database Server in Ihrer Struktur schreiben lassen.

Kann ich den Storage Provider auf programmierbare Weise steuern?

Ja, durch die SPS-Schnittstelle für den TwinCAT Storage Provider. Sie können Aufzeichnungen starten/stoppen oder historische Daten (Rohdaten oder Ergebnisdaten) abrufen.

Ist es möglich, auch Ergebnisse von der Analytics Runtime zu speichern?

Ja, das ist möglich. Hierzu können Sie beim Generieren der Analytics Runtime aus dem Analytics Workbench-Konfigurator wählen, dass die Ergebnisse an einen MQTT-Message-Broker gesendet werden sollen. Dieser Datenstrom kann vom Storage Provider erfasst werden.

Werden in TwinCAT Measurement Produkten Open Source Softwarekomponenten verwendet?

Ja es werden verschiedene Open Source Komponenten verwendet.

Bitte sehen Sie dazu die Information auf der Seite [Third-party components](#). [► 163]

Von welchen Einflussfaktoren hängt der Datendurchsatz des Storage Providers ab?

Der Datendurchsatz hängt von vielen Einflussgrößen ab. In erster Linie von System- und Netzwerkressourcen. Eine Übersicht:

- Systemeigenschaften (CPU, RAM)
- Schreibgeschwindigkeit und Qualität des Speichermediums (SSD)
- Netzwerkeigenschaften

- Komplexität der Symbolik (Datentyp, Strukturen, Arrays, etc.)
- Modus der Historisierung (Gesamte Symbolik erlaubt höheren Durchsatz, eine Teilmenge kann je nach Größe aufwendiger sein)
- Komprimierungsgrad des Streams (je stärker die Komprimierung, desto höher die Systembelastung)
- Die Größe eines Samples
- Gesamtgrößer eines Datenpakets (Anzahl Samples pro Paket)
- Die Anzahl paralleler Aufnahmen die der Storage Provider verwaltet

9.3 Third-party components

This software contains third-party components.

Please refer to the license file provided in the following folder for further information:

C:\Program Files(x86)\Beckhoff\Legal\TwinCAT-XAR-AnalyticsStorageProvider

10 Support und Service

Beckhoff und seine weltweiten Partnerfirmen bieten einen umfassenden Support und Service, der eine schnelle und kompetente Unterstützung bei allen Fragen zu Beckhoff Produkten und Systemlösungen zur Verfügung stellt.

Downloadfinder

Unser [Downloadfinder](#) beinhaltet alle Dateien, die wir Ihnen zum Herunterladen anbieten. Sie finden dort Applikationsberichte, technische Dokumentationen, technische Zeichnungen, Konfigurationsdateien und vieles mehr.

Die Downloads sind in verschiedenen Formaten erhältlich.

Beckhoff Niederlassungen und Vertretungen

Wenden Sie sich bitte an Ihre Beckhoff Niederlassung oder Ihre Vertretung für den [lokalen Support und Service](#) zu Beckhoff Produkten!

Die Adressen der weltweiten Beckhoff Niederlassungen und Vertretungen entnehmen Sie bitte unserer Internetseite: www.beckhoff.com

Dort finden Sie auch weitere Dokumentationen zu Beckhoff Komponenten.

Beckhoff Support

Der Support bietet Ihnen einen umfangreichen technischen Support, der Sie nicht nur bei dem Einsatz einzelner Beckhoff Produkte, sondern auch bei weiteren umfassenden Dienstleistungen unterstützt:

- Support
- Planung, Programmierung und Inbetriebnahme komplexer Automatisierungssysteme
- umfangreiches Schulungsprogramm für Beckhoff Systemkomponenten

Hotline: +49 5246 963-157

E-Mail: support@beckhoff.com

Beckhoff Service

Das Beckhoff Service-Center unterstützt Sie rund um den After-Sales-Service:

- Vor-Ort-Service
- Reparaturservice
- Ersatzteilservice
- Hotline-Service

Hotline: +49 5246 963-460

E-Mail: service@beckhoff.com

Beckhoff Unternehmenszentrale

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Deutschland

Telefon: +49 5246 963-0

E-Mail: info@beckhoff.com

Internet: www.beckhoff.com

Trademark statements

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.

Third-party trademark statements

Arm, Arm9 and Cortex are trademarks or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere.

FreeBSD is a registered trademark of The FreeBSD Foundation and is used by Beckhoff with the permission of The FreeBSD Foundation.

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

Mehr Informationen:
www.beckhoff.com/tf3520

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

