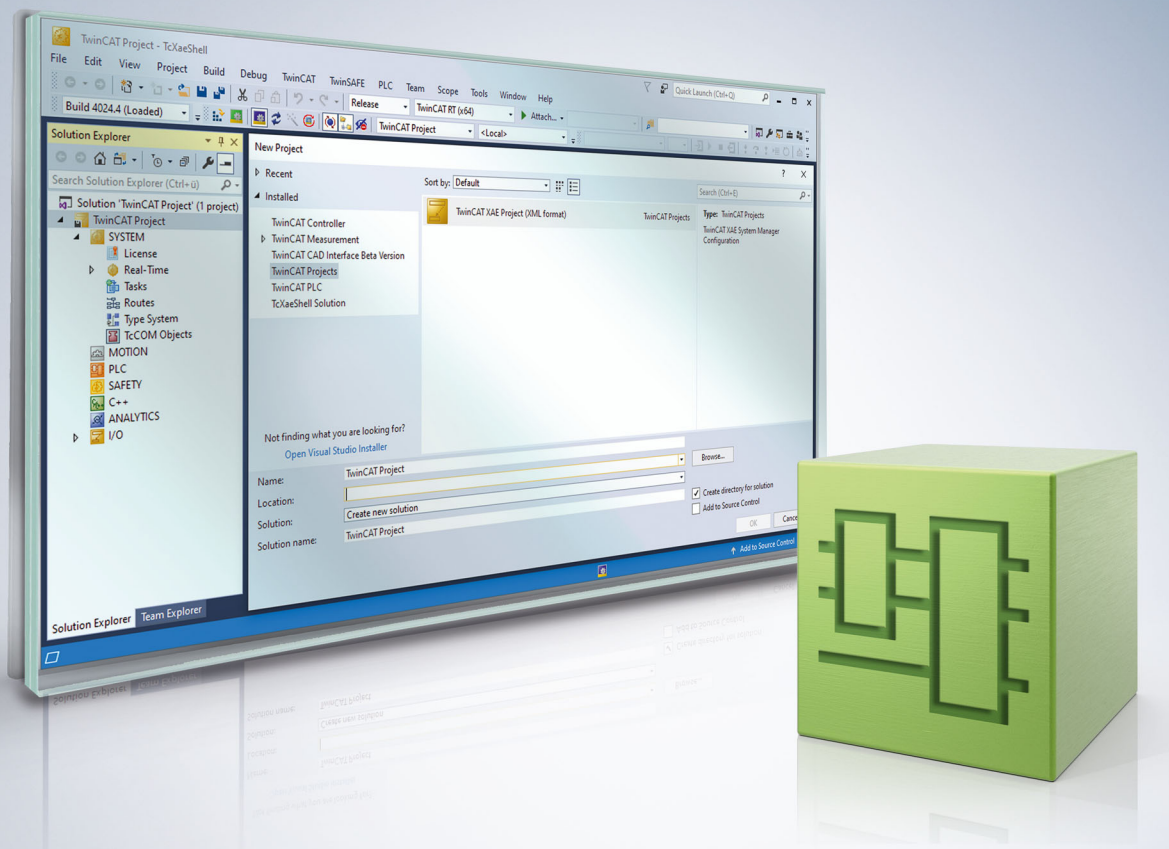
BECKHOFF New Automation Technology

Handbuch | DE

TE1000

TwinCAT 3 | PLC-Bibliothek: Tc3_EventLogger



06.12.2022 | Version: 1.5.1

Inhaltsverzeichnis

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, für jede Installation und Inbetriebnahme die zu dem betreffenden Zeitpunkt veröffentliche 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 weiter entwickelt.
Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.
Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

Marken

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.
Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Bezeichnungen führen.

Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

EtherCAT®

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

Copyright

1.2 Sicherheitshinweise

Sicherheitsbestimmungen

Beachten Sie die folgenden Sicherheitshinweise und Erklärungen!
Produktspezifische Sicherheitshinweise finden Sie auf den folgenden Seiten oder in den Bereichen Montage, Verdrahtung, Inbetriebnahme usw.

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.

Erklärung der Symbole

In der vorliegenden Dokumentation werden die folgenden Symbole mit einem nebenstehenden Sicherheitshinweis oder Hinweistext verwendet. Die Sicherheitshinweise sind aufmerksam zu lesen und unbedingt zu befolgen!

⚠ GEFAHR

Akute Verletzungsgefahr!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht unmittelbare Gefahr für Leben und Gesundheit von Personen!

⚠ WARNUNG

Verletzungsgefahr!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht Gefahr für Leben und Gesundheit von Personen!

⚠ VORSICHT

Schädigung von Personen!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, können Personen geschädigt werden!

HINWEIS

Schädigung von Umwelt oder Geräten

Wenn der Hinweis neben diesem Symbol nicht beachtet wird, können Umwelt oder Geräte geschädigt werden.

Tipp oder Fingerzeig

Dieses Symbol kennzeichnet Informationen, die zum besseren Verständnis beitragen.

1.3 Hinweise zur Informationssicherheit

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

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

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

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

2 Übersicht

Die SPS-Bibliothek Tc3_EventLogger umfasst Funktionen und Funktionsbausteine zur Verwendung des TwinCAT 3 EventLogger.

Siehe auch: Dokumentation TwinCAT 3 EventLogger

Funktionsbausteine zur Verwendung des TC3 EventLogger

FB_ListenerBase2 [▶ 26]	Basisimplementierung eines Ereignisbeobachters.
FB_TcAlarm [▶ 31]	Repräsentiert einen Alarm vom TwinCAT 3 EventLogger.
FB_TcArguments [▶ 37]	Definiert Argumente eines Ereignisses.
FB_TcEventLogger [▶ 47]	Stellt den TwinCAT 3 EventLogger selbst dar.
FB_TcMessage [▶ 56]	Repräsentiert eine Nachricht vom TwinCAT 3 EventLogger.
FB_TcSourceInfo [▶ 60]	Definiert Quellinformation eines Ereignisses.

Asynchrone Textanfragen

F_GetEventClassName [▶ 16]	Triggert die asynchrone Anfrage des Namens einer Ereignisklasse.
F_GetEventText [▶ 17]	Triggert die asynchrone Anfrage eines Ereignistextes.
FB_AsyncStrResult [▶ 11]	Ermöglicht die asynchrone Anfrage eines Textes.
FB_RequestEventClassName [▶ 12]	Ermöglicht die asynchrone Anfrage des Namens einer Ereignisklasse.
FB_RequestEventText [▶ 14]	Ermöglicht die asynchrone Anfrage eines Ereignistextes in gewünschter Sprache.

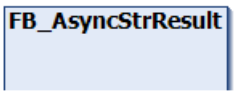
EventEntry-Konvertierung

AdsErr_TO_TcEventEntry [▶ 23]	Konvertiert einen Standard-ADS-Fehler in ein TcEventEntry.
HRESULTAdsErr_TO_TcEventEntry [▶ 24]	Konvertiert einen Standard-ADS-Fehler (HRESULT) in ein TcEventEntry.
TcEventEntry_TO_AdsErr [▶ 25]	Konvertiert ein TcEventEntry in einen Standard-ADS-Fehler.
TcEventEntry_TO_HRESULTAdsErr [▶ 25]	Konvertiert ein TcEventEntry in einen Standard-ADS-Fehler (HRESULT).

3 Funktionen und Funktionsbausteine

3.1 Asynchrone Textanfragen

3.1.1 FB_AsyncStrResult



Dieser Funktionsbaustein ermöglicht die asynchrone Anfrage eines Textes.

Syntax

Definition:

```
FUNCTION_BLOCK FB_AsyncStrResult
```

Methoden

Name	Beschreibung
GetString [▶ 11]	Sobald bBusy FALSE ist und wenn kein Fehler aufgetreten ist (bError = FALSE), kann mittels dieser Methode der angefragte Text abgeholt werden.

Eigenschaften

Name	Typ	Zugriff	Beschreibung
bBusy	BOOL	Get	TRUE, solange die Abarbeitung noch nicht abgeschlossen ist.
bError	BOOL	Get	TRUE, sobald ein Fehler eintritt.
hrErrorCode	HRESULT	Get	Gibt die Fehlerinformation aus, wenn bError TRUE ist.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.20	PC oder CX (x64, x86, ARM)	Tc3_EventLogger

3.1.1.1 GetString

Sobald bBusy = FALSE ist und wenn kein Fehler aufgetreten ist (bError = FALSE), kann mittels dieser Methode der angefragte Text abgeholt werden.

Syntax

```
METHOD GetString : BOOL
VAR_INPUT
    sResult : REFERENCE TO STRING;
    nResult : UDINT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
sResult	REFERENCE TO STRING	Puffervariable für den angefragten Text
nResult	UDINT	Puffergröße in Bytes

Rückgabewert

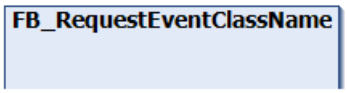
Name	Typ	Beschreibung
GetString	BOOL	Liefert TRUE, wenn der Text zugewiesen werden konnte. Liefert FALSE, wenn der Text nicht vollständig zugewiesen werden konnte, weil die angegebene Puffervariable zu klein ist.

Beispiel

Die Methode darf erst dann aufgerufen werden, wenn mittels bBusy = FALSE und bError = FALSE signalisiert wurde, dass ein Text zur Verfügung steht.

```
IF NOT fb.bBusy AND NOT fb.bError THEN
    bGetStringSuccess := fb.GetString(sText, SIZEOF(sText));
END_IF
```

3.1.2 FB_RequestEventClassName



Dieser Funktionsbaustein ermöglicht die asynchrone Anfrage des Namens einer Ereignisklasse.

Syntax

Definition:

```
FUNCTION_BLOCK FB_RequestEventClassName
```

Methoden

Name	Beschreibung
GetString [▶ 13]	Sobald bBusy FALSE ist und wenn kein Fehler aufgetreten ist (bError = FALSE), kann mittels dieser Methode der angefragte Text abgeholt werden.
Request [▶ 13]	Der Aufruf dieser Methode triggert die asynchrone Textanfrage.

Eigenschaften

Name	Typ	Zugriff	Beschreibung
bBusy	BOOL	Get	TRUE, solange die Abarbeitung noch nicht abgeschlossen ist.
bError	BOOL	Get	TRUE, sobald ein Fehler eintritt.
hrErrorCode	HRESULT	Get	Gibt die Fehlerinformation aus, wenn bError TRUE ist.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.20	PC oder CX (x64, x86, ARM)	Tc3_EventLogger

3.1.2.1 GetString

Sobald bBusy = FALSE ist und wenn kein Fehler aufgetreten ist (bError = FALSE), kann mittels dieser Methode der angefragte Text abgeholt werden.

Syntax

```
METHOD GetString : BOOL
VAR_INPUT
    sResult : REFERENCE TO STRING;
    nResult : UDINT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
sResult	REFERENCE TO STRING	Puffervariable für den angefragten Text
nResult	UDINT	Puffergröße in Bytes

Rückgabewert

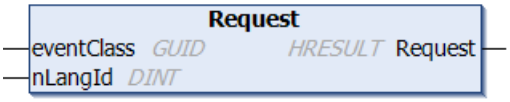
Name	Typ	Beschreibung
GetString	BOOL	Liefert TRUE, wenn der Text zugewiesen werden konnte. Liefert FALSE, wenn der Text nicht vollständig zugewiesen werden konnte, weil die angegebene Puffervariable zu klein ist.

Beispiel

Die Methode darf erst dann aufgerufen werden, wenn mittels bBusy = FALSE und bError = FALSE signalisiert wurde, dass ein Text zur Verfügung steht.

```
IF NOT fb.bBusy AND NOT fb.bError THEN
    bGetStringSuccess := fb.GetString(sText, SIZEOF(sText));
END_IF
```

3.1.2.2 Request



Der Aufruf dieser Methode triggert die asynchrone Textanfrage.

Syntax

```
METHOD Request : HRESULT
VAR_INPUT
    eventClass : GUID;
    nLangId    : DINT;
END_VAR
```

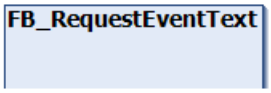
Eingänge

Name	Typ	Beschreibung
eventClass	GUID	GUID der Ereignisklasse.
nLangId	DINT	Spezifiziert die Sprach-ID Englisch (en-US) = 1033 Deutsch (de-DE) = 1031

Rückgabewert

Name	Typ	Beschreibung
Request	HRESULT	Liefert mögliche Fehlerinformationen.

3.1.3 FB_RequestEventText



Dieser Funktionsbaustein ermöglicht die asynchrone Anfrage eines Ereignistextes in gewünschter Sprache.

Syntax

Definition:

```
FUNCTION_BLOCK FB_RequestEventText
```

Methoden

Name	Beschreibung
GetString [▶ 14]	Sobald bBusy FALSE ist und wenn kein Fehler aufgetreten ist (bError = FALSE), kann mittels dieser Methode der angefragte Text abgeholt werden.
Request [▶ 15]	Der Aufruf dieser Methode triggert die asynchrone Textanfrage.

Eigenschaften

Name	Typ	Zugriff	Beschreibung
bBusy	BOOL	Get	TRUE, solange die Abarbeitung noch nicht abgeschlossen ist.
bError	BOOL	Get	TRUE, sobald ein Fehler eintritt.
hrErrorCode	HRESULT	Get	Gibt die Fehlerinformation aus, wenn bError TRUE ist.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.20	PC oder CX (x64, x86, ARM)	Tc3_EventLogger

3.1.3.1 GetString

Sobald bBusy = FALSE ist und wenn kein Fehler aufgetreten ist (bError = FALSE), kann mittels dieser Methode der angefragte Text abgeholt werden.

Syntax

```
METHOD GetString : BOOL
VAR_INPUT
    sResult : REFERENCE TO STRING;
    nResult : UDINT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
sResult	REFERENCE TO STRING	Puffervariable für den angefragten Text
nResult	UDINT	Puffergröße in Bytes

Rückgabewert

Name	Typ	Beschreibung
GetString	BOOL	Liefert TRUE, wenn der Text zugewiesen werden konnte. Liefert FALSE, wenn der Text nicht vollständig zugewiesen werden konnte, weil die angegebene Puffervariable zu klein ist.

Beispiel

Die Methode darf erst dann aufgerufen werden, wenn mittels bBusy = FALSE und bError = FALSE signalisiert wurde, dass ein Text zur Verfügung steht.

```
IF NOT fb.bBusy AND NOT fb.bError THEN
    bGetStringSuccess := fb.GetString(sText, SIZEOF(sText));
END_IF
```

3.1.3.2 Request

Der Aufruf dieser Methode triggert die asynchrone Textanfrage.

Syntax

```
METHOD Request : BOOL
VAR_INPUT
    eventClass : GUID;
    nEventId   : UDINT;
    nLangId    : DINT;
    ipArgs     : I_TcArguments;
END_VAR
```

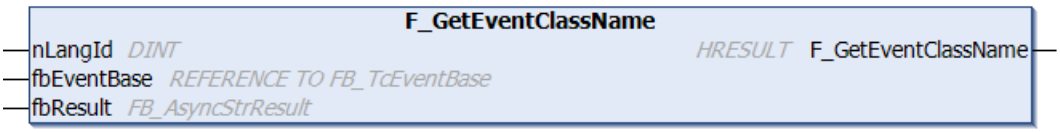
Eingänge

Name	Typ	Beschreibung
eventClass	GUID	Spezifiziert die Ereignisklasse.
nEventId	UDINT	ID von dem Ereignis.
nLangId	DINT	Spezifiziert die Sprach-ID Englisch (en-US) = 1033 Deutsch (de-DE) = 1031 …
ipArgs	I_TcArguments [▶ 63]	Optionale Angabe von Argumenten.

Rückgabewert

Name	Typ	Beschreibung
Request	HRESULT	Liefert mögliche Fehlerinformationen.

3.1.4 F_GetEventClassName



Die Funktion triggert die asynchrone Anfrage des Namens einer Ereignisklasse.

Syntax

Definition:

```
FUNCTION F_GetEventClassName : HRESULT
VAR_INPUT
    nLangId     : DINT;
    fbEventBase : REFERENCE TO FB_TcEventBase;
END_VAR
VAR_IN_OUT
    fbResult : FB_AsyncStrResult;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
nLangId	DINT	Spezifiziert die Sprach-ID Englisch (en-US) = 1033 Deutsch (de-DE) = 1031 …
fbEventBase	REFERENCE TO FB_TcEventBase [▶ 41]	Angabe eines Event/Alarm/Message-Objektes.

/ Ein-/Ausgänge

Name	Typ	Beschreibung
fbResult	FB_AsyncStrResult [▶ 11]	Angabe einer Bausteininstanz, um die asynchrone Textanfrage zu verfolgen.

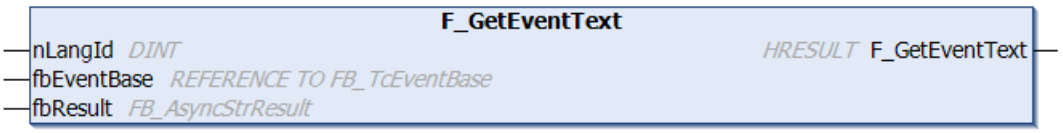
Rückgabewert

Name	Typ	Beschreibung
F_GetEventClassName	HRESULT	Liefert mögliche Fehlerinformationen.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.20	PC oder CX (x64, x86, ARM)	Tc3_EventLogger

3.1.5 F_GetEventText



Die Funktion triggert die asynchrone Anfrage eines Ereignistextes.

Syntax

Definition:

```
FUNCTION F_GetEventText : HRESULT
VAR_INPUT
    nLangId     : DINT;
    fbEventBase : REFERENCE TO FB_TcEventBase;
END_VAR
VAR_IN_OUT
    fbResult : FB_AsyncStrResult;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
nLangId	DINT	Spezifiziert die Sprach-ID Englisch (en-US) = 1033 Deutsch (de-DE) = 1031 …
fbEventBase	REFERENCE TO FB_TcEventBase [▶ 41]	Angabe eines Event/Alarm/Message-Objektes.

Ein-/Ausgänge

Name	Typ	Beschreibung
fbResult	FB_AsyncStrResult [▶ 11]	Angabe einer Bausteininstanz, um die asynchrone Textanfrage zu verfolgen.

Rückgabewert

Name	Typ	Beschreibung
F_GetEventText	HRESULT	Liefert mögliche Fehlerinformationen.

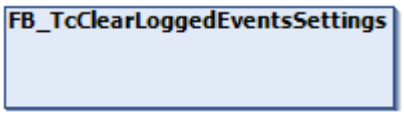
Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.20	PC oder CX (x64, x86, ARM)	Tc3_EventLogger

3.2 Filter

Die Filterfunktionalität wird an unterschiedlichen Stellen verwendet. Ein Beispiel, welches die Verwendungsmöglichkeiten beschreibt, ist das Beispiel Filter [▶ 87].

3.2.1 FB_TcClearLoggedEventsSettings



Bietet die Funktionalität, um festzulegen, welche Ereignisse aus dem Cache entfernt werden sollen.

Syntax

Definition:

```
FUNCTION_BLOCK FB_TcClearLoggedEventsSettings IMPLEMENTS I_TcClearLoggedEventsSettings
```

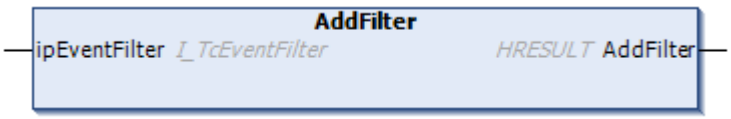
Methoden

Name	Definitionsort	Beschreibung
AddFilter	Lokal	Methode, um einen Filter hinzuzufügen. Liefert bei Erfolg `S_OK`.
Clear	Lokal	Methode zum Löschen der Einstellungen. Liefert bei Erfolg `S_OK`.
SetLimit	Lokal	Gibt die Anzahl der zu löschenden Ereignisse an. Die Begrenzung wird nach dem Sortieren und Filtern angewendet. Liefert bei Erfolg `S_OK`.
SetSorting	Lokal	Legt die Sortierung für die Abfrage fest. Liefert bei Erfolg `S_OK`.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.17	PC oder CX (x64, x86, ARM)	Tc3_EventLogger (>= v3.1.27.0)

3.2.1.1 AddFilter



Methode, um einen Filter hinzuzufügen.

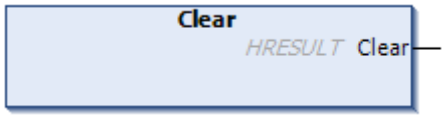
Eingänge

Name	Typ	Beschreibung
ipEventFilter	I_TcEventFilter	Instanz des zu nutzenden Filters

Rückgabewerte

Name	Typ	Beschreibung
AddFilter	HRESULT	Liefert bei Erfolg `S_OK`.

3.2.1.2 Clear



Methode zum Löschen der Einstellungen.

Rückgabewerte

Name	Typ	Beschreibung
Clear	HRESULT	Liefert bei Erfolg `S_OK`.

3.2.1.3 SetLimit

Gibt die Anzahl der zu löschenden Ereignisse an. Die Begrenzung wird nach dem Sortieren und Filtern angewendet.

Eingänge

Name	Typ	Beschreibung
eType	TcEventLimitType	Bestimmt die Referenz, ob die Begrenzung für die ersten oder letzten Ereignisse gilt.
nLimit	DINT	Spezifiziert die Anzahl (-1 = no limit)
nOffset	DINT	Optional. Definiert, wie viele Einträge übersprungen werden sollen.

Rückgabewerte

Name	Typ	Beschreibung
SetLimit	HRESULT	Liefert bei Erfolg `S_OK`.

3.2.1.4 SetSorting

Legt die Sortierung für die Abfrage fest.

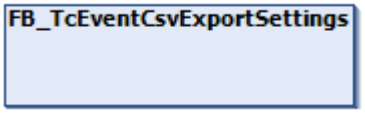
Eingänge

Name	Typ	Beschreibung
eField	TcEventField	Eigenschaft, die für die Sortierung verwendet werden soll.
eOrder	TcEventSortOrder	Definiert die Sortierreihenfolge.

Rückgabewerte

Name	Typ	Beschreibung
SetSorting	HRESULT	Liefert bei Erfolg `S_OK`.

3.2.2 FB_TcEventCsvExportSettings



Bietet die Funktionalität, den csv-Export zu spezifizieren.

Syntax

Definition:

```
FUNCTION_BLOCK FB_TcEventCsvExportSettings EXTENDS FB_TcEventExportSettings IMPLEMENTS
I_TcEventCsvExportSettings
```

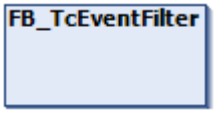
Methoden

Name	Typ	Beschreibung
bWithHeader	BOOL	Bestimmt, ob eine Kopfzeile erstellt werden soll. Standard: True
nLangId	DINT	Bestimmt die Standard-Kennung der Exportsprache. Standard: 1033
sDelimiter	STRING	Definiert das CSV-Begrenzungszeichen. Standard: Semicolon [;]

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.17	PC oder CX (x64, x86, ARM)	Tc3_EventLogger (>= v3.1.27.0)

3.2.3 FB_TcEventFilter



Stellt die Funktionalität zur Verfügung, um einen Ereignisfilter zu spezifizieren.

Die Filter werden über eine fließende Schnittstelle in Anlehnung an eine strukturierte Abfragesprache gegeben. Diese beschreibt, welche Nachrichten zutreffen sollen.

- Bedingungen können durch `.AND_OP()` und `.OR_OP()` verknüpft werden.
- Bedingungen können durch `.NOT_OP()` negiert werden.
- Bedingungen können durch Eigenschaften wie `.isAlarm()` oder beispielsweise `.EventClass.EqualsTo(<EventClass>)` definiert werden. Eine vollständige Liste der Eigenschaften befindet sich in der API-Dokumentation.
- Eine Gruppierung kann durch `.FilterExpression(<SubCondition>)` formuliert werden. Die `<SubCodition>` ist selbst wieder ein `FB_TcEventFilter` bzw. `ITcEventFilter`.

Nachdem ein Filter zusammengestellt wurde, wird er angewendet. Für das Empfangen von Nachrichten wird er beispielsweise über `FB_ListenerBase2.subscribe()` einem Empfänger zugeordnet. Der `FB_ListenerBase2` übernimmt hierdurch den Filter und gibt einen entsprechenden Rückgabewert, welcher hier beschrieben ist. Eine Änderung des Filters durch erneuten `FB_ListenerBase2.subscribe()` vornehmen.

Beispiel

Beispielsweise kann ein Filter auf diese Art zusammengestellt werden:

```
fbFilter.Severity.GreaterThan (TcEVentSeverity.Error).AND_OP().Source.Name.Like('%Main%');
```

Das Beispiel Filter [▶ 87] zeigt die Verwendung.

EtherCAT Filter

Der Empfang der EtherCAT Emergency Nachrichten ist ähnlich zu dem zuvor beschriebenen Mechanismus aufgebaut. Der Einstiegspunkt in den verketteten Methodenaufrufen ist `.EtherCATDevice()`, welches als erstes die direkte Anfrage bietet, ob es von einem EtherCAT Gerät abgesendet wurde (`IsEtherCATDevice()`). Von hier aus kann auf den Hersteller (`.VendorId()`), den ProductCode (`.ProductCode()`) sowie die Revision (`.RevisionNo()`) gefiltert werden.

Syntax

Definition:

```
FUNCTION_BLOCK FB_TcEventFilter IMPLEMENTS I_TcEventFilter, I_TcExpressionBase
```

Methoden

Name	Definitionsort	Beschreibung
Clear	I_TcEventFilter	Löscht den bisherige Filterausdruck.
FilterExpression	I_TcExpressionBase	Angabe einer unterlagerten Filterdefinition.
IsAlarm	I_TcExpressionBase	Überprüfung, ob es sich um einen Alarm handelt.
IsMessage	I_TcExpressionBase	Überprüfung, ob es sich um eine Nachricht handelt.
NOT_OP	I_TcExpressionBase	Negierung der folgenden Aussage.

Eigenschaften

Name	Typ	Zugriff	Beschreibung
AlarmState	I_TcAlarmFilterExpression	Get	Abgleich mit einem AlarmState
EtherCATDevice	I_TcEtherCATDeviceExpression	Get	Abgleich, ob es sich um ein EtherCAT Gerät als Quelle handelt.
EventClass	I_TcGuidCompare	Get	Abgleich mit einer EventClass
EventId	I_TcUDIntCompare	Get	Abgleich mit einer EventId
JsonAttribute	I_TcJsonAttributeExpression	Get	Abgleich mit dem JsonAttribut
Severity	I_TcSeverityCompare	Get	Abgleich mit der Serverity
Source	I_TcSourceInfoExpression	Get	Abgleich mit der Quelle
TimeCleared	I_TcULIntCompare	Get	Abgleich des Clear-Zeitpunkts (nur bei Alarm)
TimeConfirmed	I_TcULIntCompare	Get	Abgleich des Confirm-Zeitpunkts (nur bei Alarm mit Quittierung)
TimeRaised	I_TcULIntCompare	Get	Abgleich des Absenders (bei Nachrichten) bzw. des Raised -Zeitpunktes (bei Alarmen)

Voraussetzungen

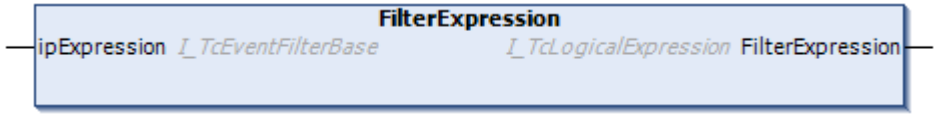
Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.17	PC oder CX (x64, x86, ARM)	Tc3_EventLogger (>= v3.1.27.0)

3.2.3.1 Clear

Rückgabewerte

Name	Typ	Beschreibung
Clear	I_TcBinaryExpression	

3.2.3.2 FilterExpression



Eingänge

Name	Typ	Beschreibung
ipExpression	I_TcEventFilterBase	Verknüpfunspunkt

Rückgabewerte

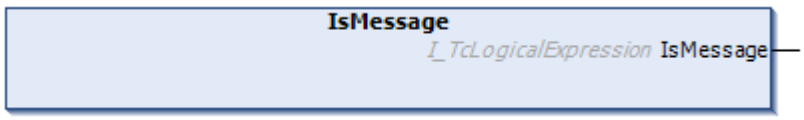
Name	Typ	Beschreibung
FilterExpression	I_TcLogicalExpression	Verknüpfunspunkt

3.2.3.3 IsAlarm

Rückgabewerte

Name	Typ	Beschreibung
IsAlarm	I_TcLogicalExpression	Verknüpfunspunkt

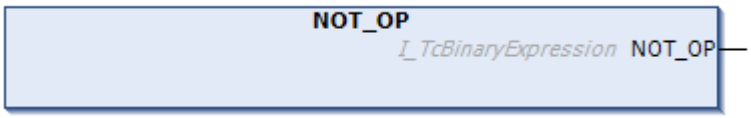
3.2.3.4 IsMessage



Rückgabewerte

Name	Typ	Beschreibung
IsMessage	I_TcLogicalExpression	Verknüpfunspunkt

3.2.3.5 NOT_OP



Rückgabewerte

Name	Typ	Beschreibung
NOT_OP	I_TcBinaryExpression	Verknüpfunspunkt

3.3 EventEntry-Konvertierung

3.3.1 AdsErr_TO_TcEventEntry

Diese Funktion konvertiert einen Standard-ADS-Fehler in ein TcEventEntry.

Syntax

Definition:

```
FUNCTION AdsErr_TO_TcEventEntry : BOOL
VAR_INPUT
    eErrorId     : E_AdsErr;
    stEventEntry : REFERENCE TO TcEventEntry;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
eErrorId	E_AdsErr	Zu konvertierender Fehlercode.
stEventEntry	REFERENCE TO TcEventEntry [▶ 80]	Gibt die resultierende Ereignisdefinition aus.

Rückgabewert

Name	Typ	Beschreibung
AdsErr_TO_TcEventEntry	BOOL	Liefert TRUE, wenn die Konvertierung erfolgreich durchgeführt wurde.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.20	PC oder CX (x64, x86, ARM)	Tc3_EventLogger

3.3.2 HRESULTAdsErr_TO_TcEventEntry

Diese Funktion konvertiert einen Standard-ADS-Fehler (HRESULT) in ein TcEventEntry.

Syntax

Definition:

```
FUNCTION HRESULTAdsErr_TO_TcEventEntry : BOOL
VAR_INPUT
    hr           : E_HRESULTAdsErr;
    stEventEntry : REFERENCE TO TcEventEntry;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
hr	E_HRESULTAdsErr	Zu konvertierender Fehlercode.
stEventEntry	REFERENCE TO TcEventEntry [▶ 80]	Gibt die resultierende Ereignisdefinition aus.

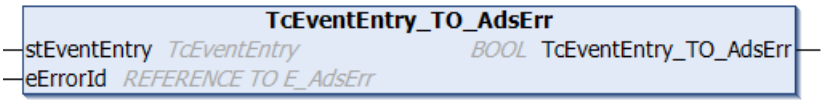
Rückgabewert

Name	Typ	Beschreibung
HRESULTAdsErr_TO_TcEventEntry	BOOL	Liefert TRUE, wenn die Konvertierung erfolgreich durchgeführt wurde. Der Aufruf schlägt fehl, wenn der Facility Code im angegebenen HRESULT unbekannt ist.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.20	PC oder CX (x64, x86, ARM)	Tc3_EventLogger

3.3.3 TcEventEntry_TO_AdsErr



Diese Funktion konvertiert ein TcEventEntry in einen Standard-ADS-Fehler.

Syntax

Definition:

```
FUNCTION TcEventEntry_TO_AdsErr : BOOL
VAR_INPUT
    stEventEntry : TcEventEntry;
    eErrorId     : REFERENCE TO E_AdsErr;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
stEventEntry	TcEventEntry [▶ 80]	Zu konvertierende Ereignisdefinition.
eErrorId	REFERENCE TO E_AdsErr	Gibt den resultierenden Fehlercode aus.

Rückgabewert

Name	Typ	Beschreibung
TcEventEntry_TO_AdsErr	BOOL	Liefert TRUE, wenn die Konvertierung erfolgreich durchgeführt wurde und FALSE, wenn die Ereignisklasse unbekannt ist.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.20	PC oder CX (x64, x86, ARM)	Tc3_EventLogger

3.3.4 TcEventEntry_TO_HRESULTAdsErr

Diese Funktion konvertiert ein TcEventEntry in einen Standard-ADS-Fehler (HRESULT).

Syntax

Definition:

```
FUNCTION TcEventEntry_TO_HRESULTAdsErr : BOOL
VAR_INPUT
    stEventEntry : TcEventEntry;
    hr           : REFERENCE TO E_HRESULTAdsErr;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
stEventEntry	TcEventEntry [▶ 80]	Zu konvertierende Ereignisdefinition.
hr	REFERENCE TO E_HRESULTAdsErr	Gibt den resultierenden Fehlercode aus.

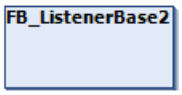
Rückgabewert

Name	Typ	Beschreibung
TcEventEntry_TO_HRESULTAdsErr	BOOL	Liefert TRUE, wenn die Konvertierung erfolgreich durchgeführt wurde und FALSE, wenn die Ereignisklassse unbekannt ist.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.20	PC oder CX (x64, x86, ARM)	Tc3_EventLogger

3.4 FB_ListenerBase2



Der Funktionsbaustein dient als Basisimplementierung eines Ereignisbeobachters.

Durch das Überschreiben der ereignisgesteuerten Methoden können neue Nachrichten und Zustandsänderungen von Alarmen erkannt werden.

Syntax

Definition:

```
FUNCTION_BLOCK FB_ListenerBase2 IMPLEMENTS I_Listener2
```

Methoden

Name	Definitionsort	Beschreibung
Execute [▶ 27]	Lokal	Muss zyklisch aufgerufen werden, damit die Ereignis-Queue abgearbeitet werden kann.
Subscribe [▶ 30]	Lokal	Meldet Benachrichtigungen an.
Unsubscribe [▶ 31]	Lokal	Meldet Benachrichtigungen ab.

Ereignisgesteuerte Methoden (Callback-Methoden)

Name	Definitionsort	Beschreibung
OnAlarmCleared [▶ 27]	I_Listener2	Wird aufgerufen, wenn der Zustand eines Alarms von „Raised“ nach „Clear“ wechselt.
OnAlarmConfirmed [▶ 28]	I_Listener2	Wird aufgerufen, wenn ein Alarm bestätigt wurde.
OnAlarmDisposed [▶ 28]	I_Listener2	Wird aufgerufen, wenn eine Alarminstanz wieder frei gegeben wurde.
OnAlarmRaised [▶ 29]	I_Listener2	Wird aufgerufen, wenn der Zustand eines Alarms von „Clear“ nach „Raised“ wechselt.
OnMessageSent [▶ 29]	I_Listener2	Wird aufgerufen, wenn eine Nachricht abgeschickt wurde.

Eigenschaften

Name	Typ	Zugriff	Definitionsort	Beschreibung
bSubscribed	BOOL	Get	Lokal	Liefert TRUE, wenn der Funktionsbaustein Benachrichtigungen angemeldet hat und die Ereignisbeobachtung aktiv ist.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.17	PC oder CX (x64, x86, ARM)	Tc3_EventLogger (>= v3.1.27.0)

3.4.1 Execute

Diese Methode muss zyklisch aufgerufen werden, damit die Ereignis-Queue abgearbeitet werden kann.

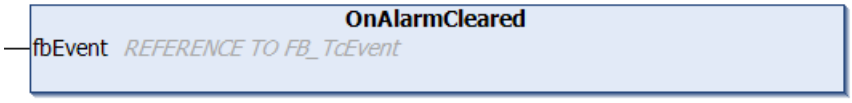
Syntax

```
METHOD Execute : HRESULT
```

Rückgabewert

Name	Typ	Beschreibung
Execute	HRESULT	Liefert S_OK, wenn der Methodenaufruf erfolgreich war, ansonsten ein HRESULT als Fehlercode

3.4.2 OnAlarmCleared



Diese Methode wird aufgerufen, wenn der Zustand eines Alarms von Raised nach Clear wechselt.

Syntax

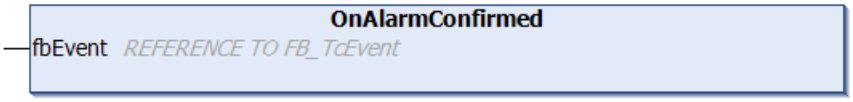
```
METHOD OnAlarmCleared : HRESULT
VAR_INPUT
    fbEvent : REFERENCE TO FB_TcEvent;
END_VAR
```

Liefert die Implementierung der Callback-Methode einen Returncode <> S_OK zurück, so werden weitere Callback-Aufrufe bis zur nächsten Ausführung pausiert.

Eingänge

Name	Typ	Beschreibung
fbEvent	REFERENCE TO FB_TcEvent [▶ 39]	Referenz auf den aufgetretenen Alarm. Diese Referenz darf nicht z.B. per Zuweisung kopiert werden.

3.4.3 OnAlarmConfirmed



Diese Methode wird aufgerufen, wenn ein Alarm bestätigt wurde.

Syntax

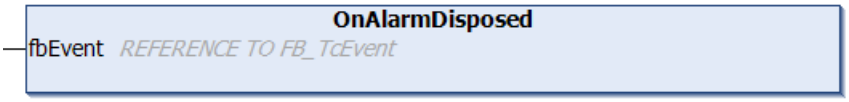
```
METHOD OnAlarmConfirmed : HRESULT
VAR_INPUT
    fbEvent : REFERENCE TO FB_TcEvent;
END_VAR
```

Liefert die Implementierung der Callback-Methode einen Returncode <> S_OK zurück, so werden weitere Callback-Aufrufe bis zur nächsten Ausführung pausiert.

Eingänge

Name	Typ	Beschreibung
fbEvent	REFERENCE TO FB_TcEvent [▶ 39]	Referenz auf den aufgetretenen Alarm. Diese Referenz darf nicht z.B. per Zuweisung kopiert werden.

3.4.4 OnAlarmDisposed



Diese Methode wird aufgerufen, wenn eine Alarminstanz wieder frei gegeben wurde.

Syntax

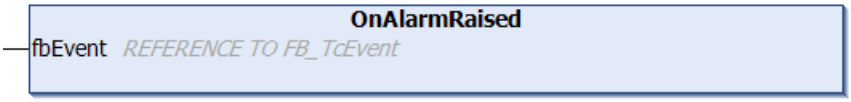
```
METHOD OnAlarmConfirmed : HRESULT
VAR_INPUT
    fbEvent : REFERENCE TO FB_TcEvent;
END_VAR
```

Liefert die Implementierung der Callback-Methode einen Returncode <> S_OK zurück, so werden weitere Callback-Aufrufe bis zur nächsten Ausführung pausiert.

Eingänge

Name	Typ	Beschreibung
fbEvent	REFERENCE TO FB_TcEvent [▶ 39]	Referenz auf den aufgetretenen Alarm. Diese Referenz darf nicht z.B. per Zuweisung kopiert werden.

3.4.5 OnAlarmRaised



Diese Methode wird aufgerufen, wenn der Zustand eines Alarms von Clear nach Raised wechselt.

Syntax

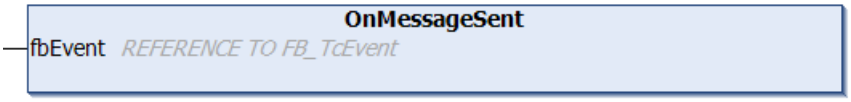
```
METHOD OnAlarmRaised : HRESULT
VAR_INPUT
    fbEvent : REFERENCE TO FB_TcEvent;
END_VAR
```

Liefert die Implementierung der Callback-Methode einen Returncode <> S_OK zurück, so werden weitere Callback-Aufrufe bis zur nächsten Ausführung pausiert.

Eingänge

Name	Typ	Beschreibung
fbEvent	REFERENCE TO FB_TcEvent [▶ 39]	Referenz auf den aufgetretenen Alarm. Diese Referenz darf nicht z.B. per Zuweisung kopiert werden.

3.4.6 OnMessageSent



Diese Methode wird aufgerufen, wenn eine Nachricht gesendet wurde.

Syntax

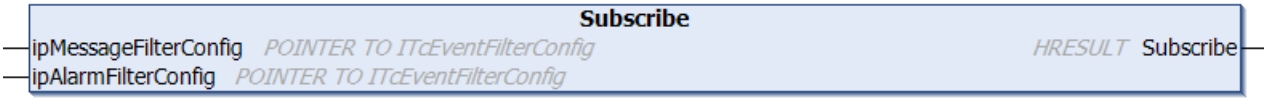
```
METHOD OnMessageSent : HRESULT
VAR_INPUT
    fbEvent : REFERENCE TO FB_TcEvent;
END_VAR
```

Liefert die Implementierung der Callback-Methode einen Returncode <> S_OK zurück, so werden weitere Callback-Aufrufe bis zur nächsten Ausführung pausiert.

Eingänge

Name	Typ	Beschreibung
fbEvent	REFERENCE TO FB_TcEvent [▶ 39]	Referenz auf das aufgetretene Ereignis. Diese Referenz darf nicht z.B. per Zuweisung kopiert werden.

3.4.7 Subscribe



Mit dieser Methode wird der Beobachter für Benachrichtigungen angemeldet.

Syntax

```
METHOD Subscribe : HRESULT
VAR_INPUT
    ipMessageFilterConfig : POINTER TO ITcEventFilterConfig;
    ipAlarmFilterConfig   : POINTER TO ITcEventFilterConfig;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
ipMessageFilterConfig	POINTER TO ITcEventFilterConfig	Zeiger auf ITcEventFilterConfig, wenn ein Filter aktiviert werden soll.
ipAlarmFilterConfig	POINTER TO ITcEventFilterConfig	Zeiger auf ITcEventFilterConfig, wenn ein Filter aktiviert werden soll.

Rückgabewert

Name	Typ	Beschreibung
Subscribe	HRESULT	Liefert S_OK, wenn der Methodenaufruf erfolgreich war. Liefert ADS_E_EXISTS, wenn der Beobachter bereits registriert ist. Liefert ansonsten ein HRESULT als Fehlercode.

3.4.8 Subscribe2

Syntax

```
METHOD Subscribe2 : HRESULT
```

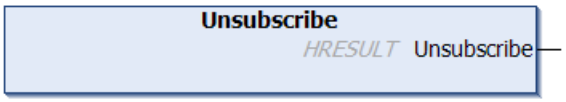
Eingang

Name	Typ	Beschreibung
ipEventFilter	I_TcEventFilterBase	Zeiger auf eine Instanz von FB_TcEventFilter [▶ 21], falls ein Filter aktiviert werden soll.

Rückgabewert

Name	Typ	Beschreibung
Subscribe2	HRESULT	Liefert bei Erfolg S_OK.

3.4.9 Unsubscribe



Mit dieser Methode wird der Beobachter abgemeldet.

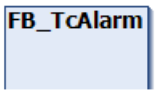
Syntax

```
METHOD Unsubscribe : HRESULT
```

Rückgabewert

Name	Typ	Beschreibung
Unsubscribe	HRESULT	Liefert S_OK, wenn der Methodenaufruf erfolgreich war. Liefert ADS_E_NOTFOUND, wenn der Beobachter nicht registriert war. Liefert ansonsten ein HRESULT als Fehlercode.

3.5 FB_TcAlarm



Dieser Funktionsbaustein repräsentiert einen Alarm des TwinCAT 3 EventLogger.

Syntax

Definition:

```
FUNCTION_BLOCK FB_TcAlarm EXTENDS FB_TcEventBase
```

Vererbungshierarchie

FB_TcEventBase [▶ 41]

FB_TcAlarm

Methoden

Name	Definitionsort	Beschreibung
EqualsTo [▶ 42]	Geerbt von FB_TcEventBase [▶ 41]	Vergleicht das Ereignis mit einer anderen Instanz.
EqualsToEventClass [▶ 43]	Geerbt von FB_TcEventBase [▶ 41]	Vergleicht die Ereignisklasse des Ereignisses mit einer anderen Ereignisklasse.
EqualsToEventEntry [▶ 43]	Geerbt von FB_TcEventBase [▶ 41]	Vergleicht die Ereignisklasse, die Event-ID und die Severity des Ereignisses mit denen eines anderen Ereignisses.
EqualsToEventEntryEx [▶ 44]	Geerbt von FB_TcEventBase [▶ 41]	Vergleicht die Ereignisdefinition des Ereignisses mit einer anderen Ereignisdefinition.
GetJsonAttribute [▶ 44]	Geerbt von FB_TcEventBase [▶ 41]	Liefert das Json-Attribut.
Release [▶ 45]	Geerbt von FB_TcEventBase [▶ 41]	Gibt die vom EventLogger erstellte Instanz wieder frei.
RequestEventClassName [▶ 45]	Geerbt von FB_TcEventBase [▶ 41]	Fragt den Namen der Ereignisklasse an.
RequestEventText [▶ 46]	Geerbt von FB_TcEventBase [▶ 41]	Liefert den Text zum Ereignis.
Clear [▶ 34]	Lokal	Setzt den Alarmzustand auf „Not-Raised“.
Confirm [▶ 34]	Lokal	Bestätigt den Alarm.
Create [▶ 35]	Lokal	Erstellt eine Alarminstanz im EventLogger.
CreateEx [▶ 36]	Lokal	Erstellt aus eine Ereignisdefinition eine Alarminstanz im EventLogger.
Raise [▶ 36]	Lokal	Setzt den Alarmzustand auf „Raised“.
SetJsonAttribute [▶ 37]	Lokal	Setzt das Json-Attribut.

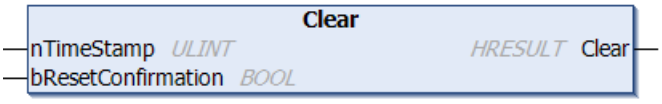
Eigenschaften

Name	Typ	Zugriff	Definitionsort	Beschreibung
eSeverity	TcEventSeverity [▶ 80]	Get	Geerbt von FB_TcEventBase [▶ 41]	Liefert die Severity.
EventClass	GUID	Get	Geerbt von FB_TcEventBase [▶ 41]	Liefert die GUID der Ereignisklasse.
ipArguments [▶ 47]	I_TcArguments [▶ 63]	Get	Geerbt von FB_TcEventBase [▶ 41]	Liefert den Schnittstellenzeiger für die Argumente.
ipSourceInfo [▶ 47]	I_TcSourceInfo [▶ 78]	Get	Geerbt von FB_TcEventBase [▶ 41]	Als Standardverhalten wird die SourceInfo intern generiert. Sie beinhaltet dann den Symbolnamen des Funktionsbausteins, der FB_TcMessage instanziiert, als SourceName sowie die Objekt-ID der SPS-Instanz als SourceID. Wenn die Instanz von FB_TcMessage mit dem Attribut 'hide' versteckt wird, kann intern kein Symbolname für das Standardverhalten generiert werden.
nEventId	nEventId	Get	Geerbt von FB_TcEventBase [▶ 41]	Liefert die ID des Ereignisses.
stEventEntry	TcEventEntry [▶ 80]	Get	Geerbt von FB_TcEventBase [▶ 41]	Liefert die Ereignisdefinition.
bRaised	BOOL	Get	Lokal	Liefert TRUE, wenn der Alarm im Zustand „Raised“ ist.
bActive	BOOL	Get	Lokal	Liefert TRUE, wenn der Alarm im Zustand „Raised“ oder „Wait For Confirmation“ ist.
eConfirmationState	TcEventConfirmationState [▶ 80]	Get	Lokal	Liefert den Bestätigungszustand.
nTimeCleared	ULINT	Get	Lokal	Liefert den Zeitpunkt des Clear.
nTimeConfirmed	ULINT	Get	Lokal	Liefert den Zeitpunkt des Confirm.
nTimeRaised	ULINT	Get	Lokal	Liefert den Zeitpunkt des Raise.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.20	PC oder CX (x64, x86, ARM)	Tc3_EventLogger

3.5.1 Clear



Diese Methode setzt den Alarmzustand auf Not-Raised.

Syntax

```
METHOD Clear : HRESULT
VAR_INPUT
    nTimeStamp         : ULINT;
    bResetConfirmation : BOOL;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
nTimeStamp	ULINT	0: Aktueller Zeitstempel wird verwendet > 0: Externer Zeitstempel in 100 Nanosekunden seit dem 1. Januar 1601 (UTC).
bResetConfirmation	BOOL	Wenn TRUE und der Bestätigungszustand WaitForConfirmation ist, wird der Bestätigungszustand auf Reset gesetzt. Ansonsten wird der Bestätigungszustand nicht verändert.

Rückgabewert

Name	Typ	Beschreibung
Clear	HRESULT	Liefert S_OK, wenn der Methodenaufruf erfolgreich war. Liefert ADS_E_INVALIDSTATE, wenn der Alarm nicht im Zustand Raised war. Liefert ansonsten ein HRESULT als Fehlercode.

3.5.2 Confirm

Setzt den Bestätigungszustand von WaitForConfirmation auf Confirmed.

Syntax

```
METHOD Confirm : HRESULT
VAR_INPUT
    nTimeStamp: ULINT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
nTimeStamp	ULINT	0: Aktueller Zeitstempel wird verwendet > 0: Externer Zeitstempel in 100 Nanosekunden seit dem 1. Januar 1601 (UTC).

Rückgabewert

Name	Typ	Beschreibung
Confirm	HRESULT	Liefert S_OK, wenn der Methodenaufruf erfolgreich war. Liefert ADS_E_INVALIDSTATE, wenn der Bestätigungszustand nicht WaitConfirmation war. Liefert ansonsten ein HRESULT als Fehlercode.

3.5.3 Create

Diese Methode erstellt eine Alarminstanz im EventLogger.

Syntax

```
METHOD Create : HRESULT
    eventClass        : GUID;
    nEventId          : UDINT;
    eSeverity         : TcEventSeverity;
    bWithConfirmation : BOOL;
    ipSourceInfo      : I_TcSourceInfo;
END_VAR
```

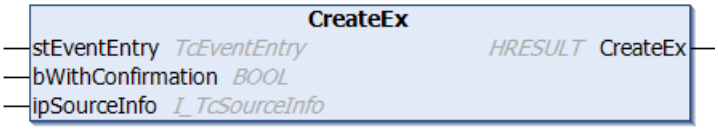
Eingänge

Name	Typ	Beschreibung
eventClass	GUID	GUID der Ereignisklasse.
nEventId	UDINT	ID des Ereignisses.
eSeverity	TcEventSeverity [▶ 80]	Severity des Ereignisses.
bWithConfirmation	BOOL	Legt fest, ob der Alarm bestätigungspflichtig ist.
ipSourceInfo	I_TcSourceInfo [▶ 78]	Schnittstellenzeiger auf die Quellinformationen. Wird kein Schnittstellenzeiger übergeben, wird eine Standardquellinformation generiert.

Rückgabewert

Name	Typ	Beschreibung
Create	HRESULT	Liefert S_OK, wenn ein neuer Alarm erfolgreich erstellt werden konnte. Liefert ERROR_ALREADY_EXISTS, wenn der Alarm bereits existiert hat. Liefert ansonsten ein HRESULT als Fehlercode

3.5.4 CreateEx



Diese Methode erstellt eine Alarminstanz im EventLogger.

Syntax

```
METHOD CreateEx : HRESULT
VAR_INPUT
    stEventEntry      : TcEventEntry;
    bWithConfirmation : BOOL;
    ipSourceInfo      : I_TcSourceInfo;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
stEventEntry	TcEventEntry [▶ 80]	Ereignisdefinition.
bWithConfirmation	BOOL	Legt fest, ob der Alarm bestätigungspflichtig ist.
ipSourceInfo	I_TcSourceInfo [▶ 78]	Schnittstellenzeiger auf die Quellinformationen. Wird kein Schnittstellenzeiger übergeben, wird eine Standardquellinformation generiert.

Rückgabewert

Name	Typ	Beschreibung
CreateEx	HRESULT	Liefert S_OK, wenn ein neuer Alarm erfolgreich erstellt werden konnte. Liefert ERROR_ALREADY_EXISTS, wenn der Alarm bereits existiert hat. Liefert ansonsten ein HRESULT als Fehlercode.

3.5.5 Raise

Setzt den Alarmzustand auf Raised.

Wenn der Alarm bestätigungspflichtig ist, wird zusätzlich der Bestätigungszustand auf WaitForConfirmation gesetzt.

Syntax

```
METHOD Raise : HRESULT
VAR_INPUT
    nTimeStamp : ULINT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
nTimeStamp	ULINT	0: Aktueller Zeitstempel wird verwendet > 0: Externer Zeitstempel in 100 Nanosekunden seit dem 1. Januar 1601 (UTC).

Rückgabewert

Name	Typ	Beschreibung
Raise	HRESULT	Liefert S_OK, wenn der Methodenaufruf erfolgreich war. Liefert ADS_E_INVALIDSTATE, wenn der Alarm bereits im Zustand Raised war. Liefert ansonsten ein HRESULT als Fehlercode

3.5.6 SetJsonAttribute

Diese Methode setzt das JSON-Attribut.

Syntax

```
METHOD SetJsonAttribute : HRESULT
VAR_IN_OUT CONSTANT
    sJsonAttribute : STRING;
END_VAR
```

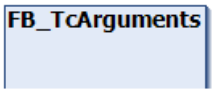
Eingänge

Name	Typ	Beschreibung
sJsonAttribute	STRING	JSON-String

Rückgabewert

Name	Typ	Beschreibung
SetJsonAttribute	HRESULT	Liefert S_OK, wenn der Methodenaufruf erfolgreich war, ansonsten ein HRESULT als Fehlercode.

3.6 FB_TcArguments



Mit diesem Funktionsbaustein können Argumente eines Ereignisses definiert werden. Er implementiert dafür die I_TcArguments-Schnittstelle.

Syntax

Definition:

```
FUNCTION_BLOCK FB_TcArguments IMPLEMENTS I_TcArguments
```

Schnittstellen

Typ	Beschreibung
I_TcArguments [▶ 63]	Definiert das Argumenten-Handling.

Methoden

Name	Definitionsort	Beschreibung
AddBlob [▶ 63]	I_TcArguments [▶ 63]	Fügt Binärdaten als Argument hinzu.
AddBool [▶ 64]	I_TcArguments [▶ 63]	Fügt ein Argument vom Typ BOOL hinzu.
AddByte [▶ 64]	I_TcArguments [▶ 63]	Fügt ein Argument vom Typ BYTE hinzu.
AddDint [▶ 65]	I_TcArguments [▶ 63]	Fügt ein Argument vom Typ DINT hinzu.
AddDWord [▶ 65]	I_TcArguments [▶ 63]	Fügt ein Argument vom Typ DWORD hinzu.
AddEventReferenceId [▶ 66]	I_TcArguments [▶ 63]	Fügt eine Referenz zu einem anderen Ereignis als Argument hinzu.
AddEventReferenceIdGuid [▶ 66]	I_TcArguments [▶ 63]	Fügt eine Referenz zu einem anderen Ereignis als Argument hinzu.
AddInt [▶ 67]	I_TcArguments [▶ 63]	Fügt ein Argument vom Typ INT hinzu.
AddLInt [▶ 67]	I_TcArguments [▶ 63]	Fügt ein Argument vom Typ LINT hinzu.
AddLReal [▶ 68]	I_TcArguments [▶ 63]	Fügt ein Argument vom Typ LREAL hinzu.
AddReal [▶ 68]	I_TcArguments [▶ 63]	Fügt ein Argument vom Typ REAL hinzu.
AddSInt [▶ 69]	I_TcArguments [▶ 63]	Fügt ein Argument vom Typ SINT hinzu.
AddString [▶ 69]	I_TcArguments [▶ 63]	Fügt ein Argument vom Typ STRING hinzu.
AddUDint [▶ 70]	I_TcArguments [▶ 63]	Fügt ein Argument vom Typ UDINT hinzu.
AddUInt [▶ 70]	I_TcArguments [▶ 63]	Fügt ein Argument vom Typ INT hinzu.
AddULInt [▶ 70]	I_TcArguments [▶ 63]	Fügt ein Argument vom Typ ULINT hinzu.
AddUSInt [▶ 71]	I_TcArguments [▶ 63]	Fügt ein Argument vom Typ USINT hinzu.
AddWord [▶ 71]	I_TcArguments [▶ 63]	Fügt ein Argument vom Typ WORD hinzu.
AddWString [▶ 72]	I_TcArguments [▶ 63]	Fügt ein Argument vom Typ WSTRING hinzu.
Clear [▶ 72]	I_TcArguments [▶ 63]	Entfernt alle Argumente.
IsEmpty [▶ 39]	Lokal	Prüft, ob Argumente hinzugefügt wurden.

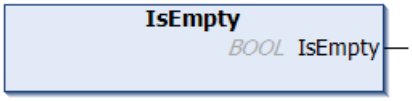
Eigenschaften

Name	Typ	Zugriff	Beschreibung
nCount	UDINT	Get	Liefert die Anzahl der übergebenen Argumente.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.20	PC oder CX (x64, x86, ARM)	Tc3_EventLogger

3.6.1 IsEmpty



Diese Methode prüft, ob Argumente hinzugefügt wurden.

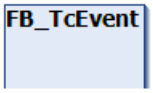
Syntax

```
METHOD IsEmpty : BOOL
```

Rückgabewert

Name	Typ	Beschreibung
IsEmpty	BOOL	Liefert TRUE, wenn keine Argumente hinzugefügt wurden.

3.7 FB_TcEvent



Dieser Funktionsbaustein stellt nur Lese-Methoden und -Eigenschaften zu einem Ereignis bereit.

Syntax

Definition:

```
FUNCTION_BLOCK FB_TcEvent EXTENDS FB_TcEventBase IMPLEMENTS I_TcEventBase
```

Vererbungshierarchie

FB_TcEventBase [▶ 41]

FB_TcEvent

Schnittstellen

Typ	Beschreibung
I_TcEventBase [▶ 73]	Basisschnittstelle, die Methoden und Eigenschaften eines Ereignisses definiert.

Methoden

Name	Definitionsort	Beschreibung
EqualsTo [▶ 42]	Geerbt von FB_TcEventBase [▶ 41]	Vergleicht das Ereignis mit einer anderen Instanz.
EqualsToEventClass [▶ 43]	Geerbt von FB_TcEventBase [▶ 41]	Vergleicht die Ereignisklasse des Ereignisses mit einer anderen Ereignisklasse.
EqualsToEventEntry [▶ 43]	Geerbt von FB_TcEventBase [▶ 41]	Vergleicht die Ereignisdefinition des Ereignisses mit einer anderen Ereignisdefinition.
EqualsToEventEntryEx [▶ 44]	Geerbt von FB_TcEventBase [▶ 41]	Vergleicht die Ereignisdefinition des Ereignisses mit einer anderen Ereignisdefinition.
GetJsonAttribute [▶ 44]	Geerbt von FB_TcEventBase [▶ 41]	Liefert das Json-Attribut.
Release [▶ 45]	Geerbt von FB_TcEventBase [▶ 41]	Gibt die vom EventLogger erstellte Instanz wieder frei.
RequestEventClassName [▶ 45]	Geerbt von FB_TcEventBase [▶ 41]	Fragt den Namen der Ereignisklasse an.
RequestEventText [▶ 46]	Geerbt von FB_TcEventBase [▶ 41]	Liefert den Text zum Ereignis.

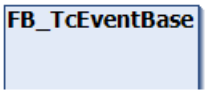
Eigenschaften

Name	Typ	Zugriff	Definitionsort	Beschreibung
eSeverity	TcEventSeverity [▶ 80]	Get	Geerbt von FB_TcEventBase [▶ 41]	Liefert die Severity.
EventClass	GUID	Get	Geerbt von FB_TcEventBase [▶ 41]	Liefert die GUID der Ereignisklasse.
ipArguments [▶ 47]	I_TcArguments [▶ 63]	Get	Geerbt von FB_TcEventBase [▶ 41]	Liefert den Schnittstellenzeiger für die Argumente.
ipSourceInfo [▶ 47]	I_TcSourceInfo [▶ 78]	Get	Geerbt von FB_TcEventBase [▶ 41]	Als Standardverhalten wird die SourceInfo intern generiert. Sie beinhaltet dann den Symbolnamen von dem Funktionsbaustein, der FB_TcMessage instanziiert, als SourceName sowie die Objekt-ID der SPS-Instanz als SourceID. Wenn die Instanz von FB_TcMessage mit dem Attribut 'hide' versteckt wird, kann intern kein Symbolname für das Standardverhalten generiert werden.
nEventId	nEventId	Get	Geerbt von FB_TcEventBase [▶ 41]	Liefert die ID des Ereignisses.
stEventEntry	TcEventEntry [▶ 80]	Get	Geerbt von FB_TcEventBase [▶ 41]	Liefert die Ereignisdefinition.
nTimestamp	ULINT	Get	Lokal	Liefert den Zeitpunkt.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.20	PC oder CX (x64, x86, ARM)	Tc3_EventLogger

3.8 FB_TcEventBase



Dieser Funktionsbaustein beinhaltet die Basisimplementierung.

Syntax

Definition:

```
FUNCTION_BLOCK FB_TcEventBase
```

Methoden

Name	Definitionsort	Beschreibung
EqualsTo [▶ 42]	Lokal	Vergleicht das Ereignis mit einer anderen Instanz.
EqualsToEventClass [▶ 43]	Lokal	Vergleicht die Ereignisklasse des Ereignisses mit einer anderen Ereignisklasse.
EqualsToEventEntry [▶ 43]	Lokal	Vergleicht die Ereignisdefinition des Ereignisses mit einer anderen Ereignisdefinition.
EqualsToEventEntryEx [▶ 44]	Lokal	Vergleicht die Ereignisdefinition des Ereignisses mit einer anderen Ereignisdefinition.
GetJsonAttribute [▶ 44]	Lokal	Liefert das Json-Attribut.
Release [▶ 45]	Lokal	Gibt die vom EventLogger erstellte Instanz wieder frei.
RequestEventClassName [▶ 45]	Lokal	Fragt den Namen der Ereignisklasse an.
RequestEventText [▶ 46]	Lokal	Liefert den Text zum Ereignis.

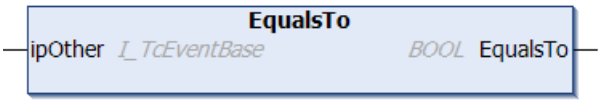
Eigenschaften

Name	Typ	Zugriff	Beschreibung
eSeverity	TcEventSeverity [▶ 80]	Get	Liefert die Severity.
EventClass	GUID	Get	Liefert die GUID der Ereignisklasse.
ipArguments [▶ 47]	I_TcArguments [▶ 63]	Get	Liefert den Schnittstellenzeiger für die Argumente.
ipSourceInfo [▶ 47]	I_TcSourceInfo [▶ 78]	Get	Als Standardverhalten wird die SourceInfo intern generiert. Sie beinhaltet dann den Symbolnamen von dem Funktionsbaustein, der FB_TcMessage instanziiert, als SourceName sowie die Objekt-ID der SPS Instanz als SourceID. Falls die Instanz von FB_TcMessage mit dem Attribut 'hide' versteckt wird, kann intern kein Symbolname für das Standardverhalten generiert werden.
nEventId	UDINT	Get	Liefert die ID des Ereignisses
nUniqueId	UDINT	Get	Liefert die Unique-ID des Ereignisses
stEventEntry	TcEventEntry [▶ 80]	Get	Liefert die Ereignisdefinition.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.20	PC oder CX (x64, x86, ARM)	Tc3_EventLogger

3.8.1 EqualsTo



Diese Methode führt einen Vergleich mit einem am Eingang angegebenen anderen Ereignis aus.

Syntax

```
METHOD EqualsTo : BOOL
VAR_INPUT
    ipOther : I_TcEventBase;
END_VAR
```

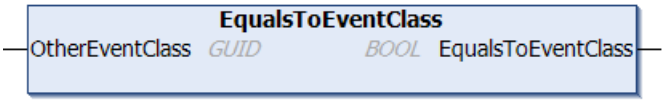
Eingänge

Name	Typ	Beschreibung
ipOther	I_TcEventBase [▶ 73]	Zu vergleichendes Ereignis

Rückgabewert

Name	Typ	Beschreibung
EqualsTo	BOOL	Liefert TRUE, wenn die Ereignisse übereinstimmen.

3.8.2 EqualsToEventClass



Diese Methode führt einen Vergleich mit einer am Eingang angegebenen anderen Ereignisklasse aus.

Syntax

```
METHOD EqualsToEventClass : BOOL
VAR_INPUT
    OtherEventClass : GUID
END_VAR
```

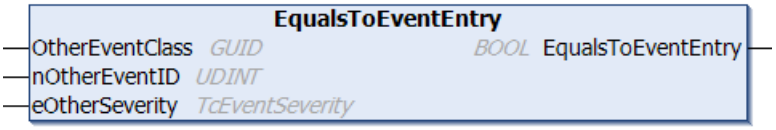
Eingänge

Name	Typ	Beschreibung
OtherEventClass	GUID	Zu vergleichende Ereignisklasse.

Rückgabewert

Name	Typ	Beschreibung
EqualsToEventClass	BOOL	Liefert TRUE, wenn die Ereignisklassen übereinstimmen.

3.8.3 EqualsToEventEntry



Diese Methode führt einen Vergleich mit einem am Eingang angegebenen anderen Ereignis aus.

Syntax

```
METHOD EqualsToEventEntry : BOOL
VAR_INPUT
    OtherEventClass : GUID;
    nOtherEventID   : UDINT;
    eOtherSeverity  : TcEventSeverity;
END_VAR
```

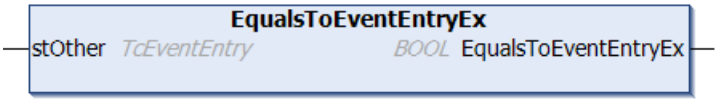
Eingänge

Name	Typ	Beschreibung
OtherEventClass	GUID	Ereignisklasse des zu vergleichenden Ereignisses.
nOtherEventID	UDINT	Event-ID des zu vergleichenden Ereignisses.
eOtherSeverity	TcEventSeverity [▶ 80]	Event-Severity des zu vergleichenden Ereignisses.

Rückgabewert

Name	Typ	Beschreibung
EqualsToEventEntry	BOOL	Liefert TRUE, wenn die Ereignisse übereinstimmen.

3.8.4 EqualsToEventEntryEx



Diese Methode führt einen Vergleich mit einem am Eingang angegebenen anderen Ereignis aus.

Syntax

```
METHOD EqualsToEventEntryEx : BOOL
VAR_INPUT
    stOther : TcEventEntry;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
stOther	TcEventEntry [▶ 80]	Zu vergleichendes Ereignis.

Rückgabewert

Name	Typ	Beschreibung
EqualsToEventEntryEx	BOOL	Liefert TRUE, wenn die Ereignisse übereinstimmen.

3.8.5 GetJsonAttribute

Diese Methode liefert das JSON-Attribut.

Syntax

```
METHOD GetJsonAttribute : HRESULT
VAR_INPUT
    sJsonAttribute : REFERENCE TO STRING;
    nJsonAttribute : UDINT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
sJsonAttribute	REFERENCE TO STRING	Referenz auf eine Variable vom Typ String
nJsonAttribute	UDINT	Länge der String-Variable

Rückgabewert

Name	Typ	Beschreibung
GetJsonAttribute	HRESULT	Liefert S_OK, wenn der Methodenaufruf erfolgreich war. Liefert ERROR_BAD_LENGTH, wenn die Länge der Variable zu klein ist. Ansonsten wird HRESULT als Fehlercode zurückgegeben.

3.8.6 Release

Diese Methode gibt die vom Eventlogger erstellte Instanz wieder frei.

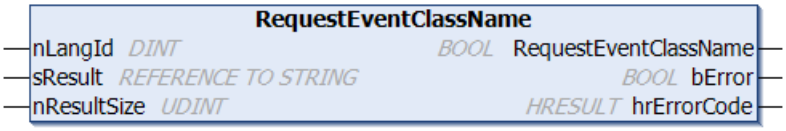
Syntax

```
METHOD Release : HRESULT
```

Rückgabewert

Name	Typ	Beschreibung
Release	HRESULT	Liefert S_OK, wenn der Methodenaufruf erfolgreich war, ansonsten ein HRESULT als Fehlercode.

3.8.7 RequestEventClassName



Diese Methode liefert den Namen der Ereignisklasse.

Syntax

```
METHOD RequestEventClassName : BOOL
VAR_INPUT
    nLangId     : DINT;
    sResult     : REFERENCE TO STRING;
    nResultSize : UDINT;
END_VAR
VAR_OUTPUT
    bError      : BOOL;
    hrErrorCode : HRESULT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
nLangId	DINT	Spezifiziert die Sprach-ID Englisch (en-US) = 1033 Deutsch (de-DE) = 1031 …
sResult	REFERENCE TO STRING	Referenz auf eine Variable vom Typ String
nResultSize	UDINT	Größe der String-Variable in Bytes

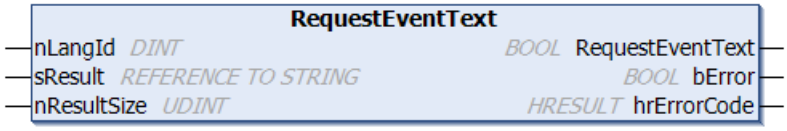
Rückgabewert

Name	Typ	Beschreibung
RequestEventClassName	BOOL	Liefert TRUE, sobald die Abfrage beendet wurde. Wenn die asynchrone Abfrage noch aktiv ist, liefert der Rückgabewert FALSE. Die Methode muss so lange aufgerufen werden, bis der Rückgabewert TRUE ist.

Ausgänge

Name	Typ	Beschreibung
bError	BOOL	Liefert FALSE, wenn der Methodenaufruf erfolgreich war. Liefert TRUE, wenn ein Fehler aufgetreten ist.
hrErrorCode	HRESULT	Liefert S_OK, wenn der Methodenaufruf erfolgreich war. Im Falle eines Fehlers wird ein Fehlercode ausgegeben.

3.8.8 RequestEventText



Diese Methode liefert den Ereignistext.

Syntax

```
METHOD RequestEventText : BOOL
VAR_INPUT
    nLangId     : DINT;
    sResult     : REFERENCE TO STRING;
    nResultSize : UDINT;
END_VAR
VAR_OUTPUT
    bError      : BOOL;
    hrErrorCode : HRESULT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
nLangId	DINT	Spezifiziert die Sprach-ID Englisch (en-US) = 1033 Deutsch (de-DE) = 1031 …
sResult	REFERENCE TO STRING	Referenz auf eine Variable vom Typ String
nResultSize	UDINT	Größe der String-Variable in Bytes

Rückgabewert

Name	Typ	Beschreibung
RequestEventText	BOOL	Liefert TRUE, sobald die Abfrage beendet wurde. Wenn die asynchrone Abfrage noch aktiv ist, liefert der Rückgabewert FALSE. Die Methode muss so lange aufgerufen werden, bis der Rückgabewert TRUE ist.

Ausgänge

Name	Typ	Beschreibung
bError	BOOL	Liefert FALSE, wenn der Methodenaufruf erfolgreich war. Liefert TRUE, wenn ein Fehler aufgetreten ist.
hrErrorCode	HRESULT	Liefert S_OK, wenn der Methodenaufruf erfolgreich war. Im Falle eines Fehlers wird ein Fehlercode ausgegeben.

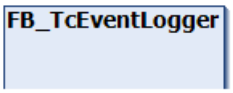
3.8.9 ipArguments

```
PROPERTY PUBLIC ipArguments : I_TcArguments
```

3.8.10 ipSourceInfo

```
PROPERTY ipSourceInfo : I_TcSourceInfo
```

3.9 FB_TcEventLogger



Dieser Funktionsbaustein stellt den TwinCAT 3 EventLogger selbst dar.

Syntax

Definition:

```
FUNCTION_BLOCK FB_TcEventLogger
```

Methoden

Name	Beschreibung
ClearAlarms [▶ 48]	Löscht aktive Alarme.
ClearAllAlarms [▶ 49]	Ruft Clear() für alle Alarme im Zustand Raised auf.
ClearLoggedEvents [▶ 49]	Löscht protokollierte Ereignisse.
ConfirmAlarms [▶ 50]	
ConfirmAllAlarms [▶ 50]	Ruft Confirm() für alle Alarme mit dem Bestätigungszustand WaitForConfirmation auf.
ExportLoggedEvents [▶ 51]	Exportiert protokollierte Ereignisse.
GetAlarm [▶ 51]	Liefert einen Zeiger auf einen existierenden Alarm.
GetAlarmEx [▶ 52]	Liefert einen Zeiger auf einen existierenden Alarm.
IsAlarmRaised [▶ 52]	Fragt ab, ob ein Alarm im Zustand Raised ist.
IsAlarmRaisedEx [▶ 53]	Fragt ab, ob ein Alarm im Zustand Raised ist.
SendMessage [▶ 54]	Sendet eine Nachricht.
SendMessage2 [▶ 54]	Sendet eine Nachricht.
SendMessageEx [▶ 55]	Sendet eine Nachricht.
SendMessageEx2 [▶ 56]	Sendet eine Nachricht.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.20	PC oder CX (x64, x86, ARM)	Tc3_EventLogger

3.9.1 ClearAlarms

Methode zum Löschen aktiver Alarme. Gibt `S_OK` zurück, wenn erfolgreich.

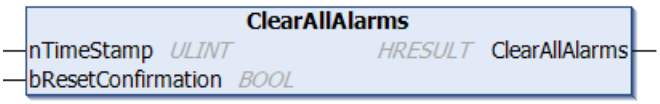
Eingänge

Name	Typ	Beschreibung
nTimeStamp	ULINT	0 setzen, um die aktuelle Zeit automatisch zu erhalten. Initial: 0
bResetConfirmation	BOOL	Wenn TRUE und der Bestätigungsstatus WaitForConfirmation ist, wird der Bestätigungsstatus auf Reset gesetzt. Andernfalls wird der Bestätigungsstatus nicht geändert. Initial: FALSE
ipFilter	I_TcEventFilter	Angeben, welche Alarme gelöscht werden sollen, ansonsten werden alle ausgelösten Alarme gelöscht.

Rückgabewerte

Name	Typ	Beschreibung
ClearAlarms	HRESULT	

3.9.2 ClearAllAlarms



Diese Methode ruft für alle Alarme im Alarmzustand Raised deren Methode Clear() auf.

Syntax

```
METHOD ClearAllAlarms : HRESULT
VAR_INPUT
    nTimeStamp         : ULINT := 0;
    bResetConfirmation : BOOL := FALSE;
END_VAR
```

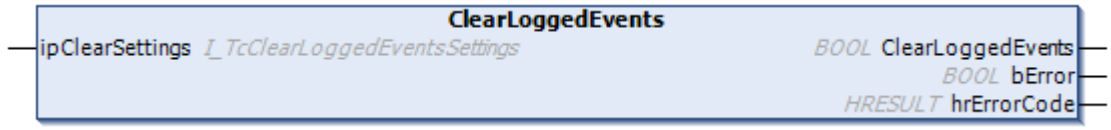
Eingänge

Name	Typ	Beschreibung
nTimeStamp	ULINT	0: Aktueller Zeitstempel wird verwendet > 0: Externer Zeitstempel in 100 Nanosekunden seit dem 1. Januar 1601 (UTC).
bResetConfirmation	BOOL	Wenn TRUE und der Bestätigungszustand WaitForConfirmation ist, wird der Bestätigungszustand auf Reset gesetzt. Ansonsten wird der Bestätigungszustand nicht verändert.

Rückgabewert

Name	Typ	Beschreibung
ClearAllAlarms	HRESULT	Liefert S_OK, wenn der Methodenaufruf erfolgreich war, ansonsten ein HRESULT als Fehlercode

3.9.3 ClearLoggedEvents



Async-Methode zum Löschen von protokollierten Ereignissen. Gibt TRUE zurück, wenn die asynchrone Anfrage nicht mehr belegt ist.

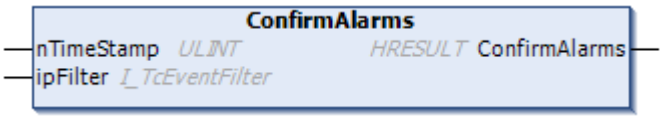
Eingänge

Name	Typ	Beschreibung
ipClearSettings	I_TcClearLoggedEventsSettings	Optional (andernfalls wird der ganze Cache geleert)

Rückgabewerte

Name	Typ	Beschreibung
ClearLoggedEvents	BOOL	
bError	BOOL	
hrErrorCode	HRESULT	

3.9.4 ConfirmAlarms



Eingänge

Name	Typ	Beschreibung
nTimeStamp	ULINT	0 setzen, um die aktuelle Zeit automatisch zu erhalten. Initial: 0
ipFilter	I_TcEventFilter	Festlegen, welche Alarme bestätigt werden sollen, ansonsten werden alle Alarme mit dem Bestätigungsstatus WaitForConfirmation bestätigt.

Rückgabewerte

Name	Typ	Beschreibung
ConfirmAlarms	HRESULT	

3.9.5 ConfirmAllAlarms

Diese Methode ruft für alle Alarme mit dem Bestätigungszustand WaitForConfirmation deren Methode Confirm() auf.

Syntax

```
METHOD ConfirmAllAlarms : HRESULT
VAR_INPUT
    nTimeStamp : ULINT := 0;
END_VAR
```

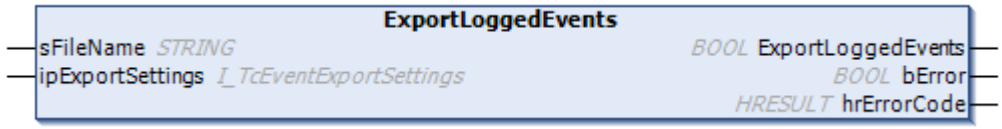
Eingänge

Name	Typ	Beschreibung
nTimeStamp	ULINT	0: Aktueller Zeitstempel wird verwendet > 0: Externer Zeitstempel in 100 Nanosekunden seit dem 1. Januar 1601 (UTC).

Rückgabewert

Name	Typ	Beschreibung
ConfirmAllAlarms	HRESULT	Liefert S_OK, wenn der Methodenaufruf erfolgreich war, ansonsten ein HRESULT als Fehlercode.

3.9.6 ExportLoggedEvents



Exportiert protokollierte Ereignisse asynchron. Gibt TRUE zurück, wenn die asynchrone Abarbeitung abgeschlossen ist.

Eingänge

Name	Typ	Beschreibung
sFileName	STRING	Name der Zieldatei
ipExportSettings	I_TcEventExportSettings	Angeben, welche Ereignisse exportiert werden sollen, ansonsten werden alle Ereignisse exportiert. Hierfür kann eine Instanz von FB_TcEventCsvExportSettings [▶ 20] zugewiesen werden.

Rückgabewerte

Name	Typ	Beschreibung
ExportLoggedEvents	BOOL	TRUE, wenn die Abarbeitung abgeschlossen ist.
bError	BOOL	TRUE, sobald ein Fehler eintritt.
hrErrorCode	HRESULT	Gibt die Fehlerinformation aus, wenn `bError` TRUE ist.

3.9.7 GetAlarm

Liefert einen Schnittstellenzeiger auf eine existierende Instanz.

Syntax

```
METHOD GetAlarm : HRESULT
VAR_INPUT
    eventClass   : GUID;
    nEventId     : UDINT;
    ipSourceInfo : I_TcSourceInfo := 0;
    fbAlarm      : REFERENCE TO FB_TcAlarm;
END_VAR
```

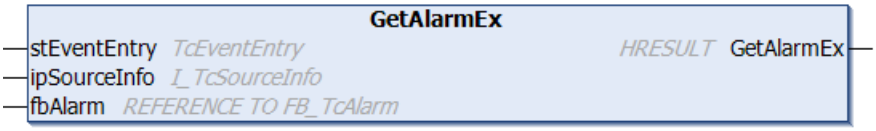
Eingänge

Name	Typ	Beschreibung
eventClass	GUID	GUID der Ereignisklasse.
nEventId	UDINT	ID des Ereignisses
ipSourceInfo	I_TcSourceInfo [▶ 78]	Zeiger auf eine ITcSourceInfo-Schnittstelle.
fbAlarm	REFERENCE TO FB_TcAlarm [▶ 31]	Zeiger auf einen Alarm.

Rückgabewert

Name	Typ	Beschreibung
GetAlarm	HRESULT	Liefet ADS_E_NOTFOUND, wenn keine Instanz gefunden wurde. Liefert S_OK, wenn der Methodenaufruf erfolgreich war, ansonsten ein HRESULT als Fehlercode.

3.9.8 GetAlarmEx



Liefert einen Schnittstellenzeiger auf eine existierende Instanz.

Syntax

```
METHOD GetAlarmEx : HRESULT
VAR_INPUT
    stEventEntry : TcEventEntry;
    ipSourceInfo : I_TcSourceInfo := 0; // optional
    fbAlarm      : REFERENCE TO FB_TcAlarm;
END_VAR
```

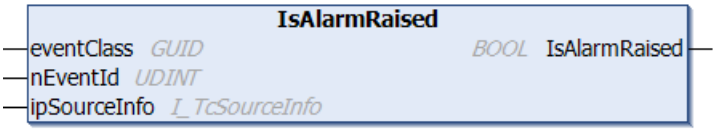
Eingänge

Name	Typ	Beschreibung
stEventEntry	TcEventEntry [▶ 80]	Ereignisdefinition.
ipSourceInfo	I_TcSourceInfo [▶ 78]	Zeiger auf eine ITcSourceInfo-Schnittstelle.
fbAlarm	REFERENCE TO FB_TcAlarm [▶ 31]	Zeiger auf einen Alarm.

Rückgabewert

Name	Typ	Beschreibung
GetAlarmEx	HRESULT	Liefet ADS_E_NOTFOUND, wenn keine Instanz gefunden wurde. Liefert S_OK, wenn der Methodenaufruf erfolgreich war, ansonsten ein HRESULT als Fehlercode.

3.9.9 IsAlarmRaised



Diese Methode fragt ab, ob ein Alarm im Zustand Raised ist.

Syntax

```
METHOD IsAlarmRaised : BOOL
VAR_INPUT
    eventClass   : GUID;
    nEventId     : UDINT;
    ipSourceInfo : I_TcSourceInfo := 0;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
eventClass	GUID	GUID der Ereignisklasse.
nEventId	UDINT	ID des Ereignisses.
ipSourceInfo	I_TcSourceInfo [▶ 78]	Zeiger auf eine ITcSourceInfo-Schnittstelle.

Rückgabewert

Name	Typ	Beschreibung
IsAlarmRaised	BOOL	Liefert S_OK, wenn der Methodenaufruf erfolgreich war, ansonsten ein HRESULT als Fehlercode.

3.9.10 IsAlarmRaisedEx

Diese Methode fragt ab, ob ein Alarm im Zustand Raised ist.

Syntax

```
METHOD IsAlarmRaisedEx : BOOL
VAR_INPUT
    stEventEntry : TcEventEntry;
    ipSourceInfo : I_TcSourceInfo := 0;
END_VAR
```

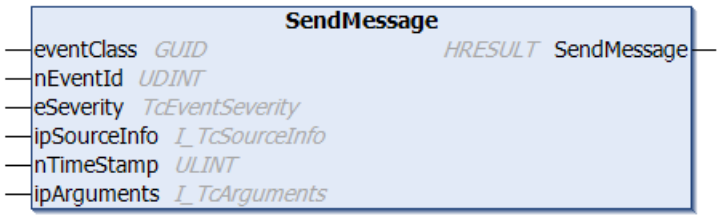
Eingänge

Name	Typ	Beschreibung
stEventEntry	UDINT	Ereignisdefinition.
ipSourceInfo	I_TcSourceInfo [▶ 78]	Zeiger auf eine ITcSourceInfo-Schnittstelle.

Rückgabewert

Name	Typ	Beschreibung
IsAlarmRaisedEx	BOOL	Liefert TRUE, wenn der Alarm im Zustand Raised ist.

3.9.11 SendMessage



Diese Methode sendet eine Nachricht.

Syntax

```
METHOD SendMessage : HRESULT
VAR_INPUT
    eventClass    : GUID;
    nEventId      : UDINT;
    eSeverity     : TcEventSeverity;
    ipSourceInfo  : I_TcSourceInfo := 0;
    nTimeStamp    : ULINT := 0;
    ipArguments   : I_TcArguments := 0;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
eventClass	GUID	GUID der Ereignisklasse.
nEventId	UDINT	ID des Ereignisses.
eSeverity	TcEventSeverity [▶ 80]	Severity des Ereignisses.
ipSourceInfo	I_TcSourceInfo [▶ 78]	Zeiger auf eine ITcSourceInfo-Schnittstelle.
nTimeStamp	ULINT	0: Aktueller Zeitstempel wird verwendet. > 0: Externer Zeitstempel in 100 Nanosekunden seit dem 1. Januar 1601 (UTC).
ipArguments	I_TcArguments [▶ 63]	Zeiger auf ITcArguments-Schnittstelle.

Rückgabewert

Name	Typ	Beschreibung
SendMessage	HRESULT	Liefert S_OK, wenn der Methodenaufruf erfolgreich war, ansonsten ein HRESULT als Fehlercode.

3.9.12 SendMessage2

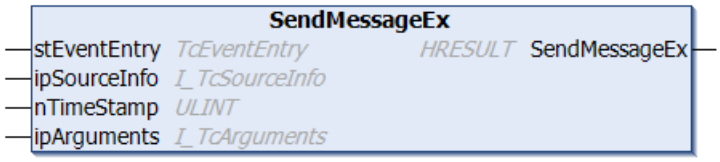
Eingänge

Name	Typ	Beschreibung
eventClass	GUID	
nEventId	UDINT	
eSeverity	TcEventSeverity	
ipSourceInfo	I_TcSourceInfo	Optional Initial: 0
nTimeStamp	ULINT	Setzen Sie 0, um die aktuelle Zeit automatisch zu erhalten. Initial: 0
ipArguments	I_TcArguments	Optional Initial: 0
sJsonAttribute	STRING	

Rückgabewerte

Name	Typ	Beschreibung
SendMessage2	HRESULT	

3.9.13 SendMessageEx



Diese Methode sendet eine Nachricht.

Syntax

```
METHOD SendMessageEx : HRESULT
VAR_INPUT
    stEventEntry : TcEventEntry;
    ipSourceInfo : I_TcSourceInfo := 0;
    nTimeStamp   : ULINT := 0;
    ipArguments  : I_TcArguments := 0;
END_VAR
```

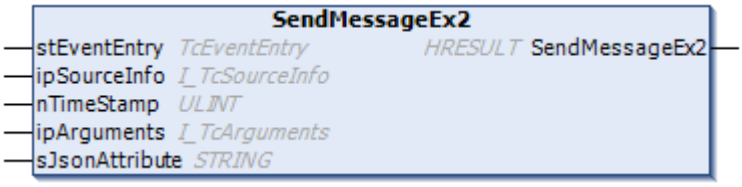
Eingänge

Name	Typ	Beschreibung
stEventEntry	TcEventEntry [▶ 80]	Ereignisdefinition.
ipSourceInfo	I_TcSourceInfo [▶ 78]	Zeiger auf eine ITcSourceInfo-Schnittstelle.
nTimeStamp	ULINT	0: Aktueller Zeitstempel wird verwendet > 0: Externer Zeitstempel in 100 Nanosekunden seit dem 1. Januar 1601 (UTC).
ipArguments	I_TcArguments [▶ 63]	Zeiger auf ITcArguments-Schnittstelle.

Rückgabewert

Name	Typ	Beschreibung
SendMessageEx	HRESULT	Liefert S_OK, wenn der Methodenaufruf erfolgreich war, ansonsten ein HRESULT als Fehlercode.

3.9.14 SendMessageEx2



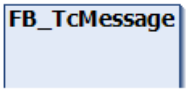
Eingänge

Name	Typ	Beschreibung
stEventEntry	TcEventEntry	
ipSourceInfo	I_TcSourceInfo	Optional Initial: 0
nTimeStamp	ULINT	0 setzen, um die aktuelle Zeit automatisch zu erhalten. Initial: 0
ipArguments	I_TcArguments	Optional Initial: 0
sJsonAttribute	STRING	

Rückgabewerte

Name	Typ	Beschreibung
SendMessageEx2	HRESULT	

3.10 FB_TcMessage



Dieser Funktionsbaustein repräsentiert eine Nachricht vom TwinCAT 3 EventLogger.

Syntax

Definition:

```
FUNCTION_BLOCK FB_TcMessage EXTENDS FB_TcEventBase IMPLEMENTS I_TcMessage
```

Vererbungshierarchie

FB_TcEventBase [▶ 41]

FB_TcMessage

Schnittstellen

Typ	Beschreibung
I_TcMessage [▶ 78]	Stellt Methoden und Eigenschaften für das Nachrichten-Handling bereit.

Methoden

Name	Definitionsort	Beschreibung
EqualsTo [▶ 42]	Geerbt von FB_TcEventBase [▶ 41]	Vergleicht das Ereignis mit einer anderen Instanz.
EqualsToEventClass [▶ 43]	Geerbt von FB_TcEventBase [▶ 41]	Vergleicht die Ereignisklasse des Ereignisses mit einer anderen Ereignisklasse.
EqualsToEventEntry [▶ 43]	Geerbt von FB_TcEventBase [▶ 41]	Vergleicht die Ereignisdefinition des Ereignisses mit einer anderen Ereignisdefinition.
EqualsToEventEntryEx [▶ 44]	Geerbt von FB_TcEventBase [▶ 41]	Vergleicht die Ereignisdefinition des Ereignisses mit einer anderen Ereignisdefinition.
GetJsonAttribute [▶ 44]	Geerbt von FB_TcEventBase [▶ 41]	Liefert das Json-Attribut.
Release [▶ 45]	Geerbt von FB_TcEventBase [▶ 41]	Gibt die vom EventLogger erstellte Instanz wieder frei.
RequestEventClassName [▶ 45]	Geerbt von FB_TcEventBase [▶ 41]	Fragt den Namen der Ereignisklasse an.
RequestEventText [▶ 46]	Geerbt von FB_TcEventBase [▶ 41]	Liefert den Text zum Ereignis.
Create [▶ 58]	Lokal	Erstellt eine Nachrichtinstanz im EventLogger.
CreateEx [▶ 59]	Lokal	Erstellt aus eine Ereignisdefinition eine Nachrichtinstanz im EventLogger.
SetJsonAttribute [▶ 59]	Lokal	Setzt das Json-Attribut.
Send [▶ 78]	I_TcMessage [▶ 78]	Sendet eine Nachricht.

Eigenschaften

Name	Typ	Zugriff	Definitionsort	Beschreibung
eSeverity	TcEventSeverity [▶ 80]	Get	Geerbt von FB_TcEventBase [▶ 41]	Liefert die Severity.
EventClass	GUID	Get	Geerbt von FB_TcEventBase [▶ 41]	Liefert die GUID der Ereignisklasse.
ipArguments [▶ 47]	I_TcArguments [▶ 63]	Get	Geerbt von FB_TcEventBase [▶ 41]	Liefert den Schnittstellenzeiger für die Argumente.
ipSourceInfo [▶ 47]	I_TcSourceInfo [▶ 78]	Get	Geerbt von FB_TcEventBase [▶ 41]	Als Standardverhalten wird die SourceInfo intern generiert. Sie beinhaltet dann den Symbolnamen von dem Funktionsbaustein, der FB_TcMessage instanziiert, als SourceName sowie die Objekt-ID der SPS-Instanz als SourceID. Wenn die Instanz von FB_TcMessage mit dem Attribut 'hide' versteckt wird, kann intern kein Symbolname für das Standardverhalten generiert werden.
nEventId	nEventId	Get	Geerbt von FB_TcEventBase [▶ 41]	Liefert die ID des Ereignisses.
stEventEntry	TcEventEntry [▶ 80]	Get	Geerbt von FB_TcEventBase [▶ 41]	Liefert die Ereignisdefinition.
nTimeSent	ULINT	Get	Lokal	Liefer den Zeitpunkt des Send.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.20	PC oder CX (x64, x86, ARM)	Tc3_EventLogger

3.10.1 Create

Diese Methode erstellt eine Nachrichtinstanz im EventLogger.

Syntax

```
METHOD Create : HRESULT
VAR_INPUT
    eventClass   : GUID;
    nEventId     : UDINT;
    eSeverity    : TcEventSeverity;
    ipSourceInfo : I_TcSourceInfo := 0;
END_VAR
```

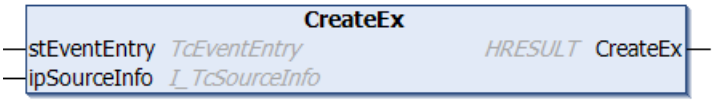
Eingänge

Name	Typ	Beschreibung
eventClass	GUID	GUID der Ereignisklasse.
nEventId	UDINT	ID des Ereignisses.
eSeverity	TcEventSeverity [▶ 80]	Definiert die Severity.
ipSourceInfo	I_TcSourceInfo [▶ 78]	Zeiger auf eine ITcSourceInfo-Schnittstelle.

Rückgabewert

Name	Typ	Beschreibung
Create	HRESULT	Liefert S_OK, wenn der Methodenaufruf erfolgreich war, ansonsten ein HRESULT als Fehlercode.

3.10.2 CreateEx



Diese Methode erstellt aus eine Ereignisdefinition eine Nachrichtinstanz im EventLogger.

Syntax

```
METHOD PUBLIC CreateEx : HRESULT
VAR_INPUT
    stEventEntry : TcEventEntry;
    ipSourceInfo : I_TcSourceInfo := 0;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
stEventEntry	TcEventEntry [▶ 80]	Ereignisdefinition.
ipSourceInfo	I_TcSourceInfo [▶ 78]	Schnittstellenzeiger auf die Quellinformationen. Wird kein Schnittstellenzeiger übergeben, wird eine Standardquellinformation generiert.

Rückgabewert

Name	Typ	Beschreibung
CreateEx	HRESULT	Liefert S_OK, wenn der Methodenaufruf erfolgreich war, ansonsten ein HRESULT als Fehlercode.

3.10.3 SetJsonAttribute

Diese Methode setzt das JSON-Attribut.

Syntax

```
METHOD SetJsonAttribute : HRESULT
VAR_IN_OUT CONSTANT
    sJsonAttribute : STRING;
END_VAR
```

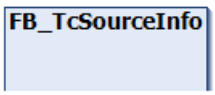
Eingänge

Name	Typ	Beschreibung
sJsonAttribute	STRING	JSON-String

Rückgabewert

Name	Typ	Beschreibung
SetJsonAttribute	HRESULT	Liefert S_OK, wenn der Methodenaufruf erfolgreich war, ansonsten ein HRESULT als Fehlercode.

3.11 FB_TcSourceInfo



Mit diesem Funktionsbaustein kann die Quellinformation eines Ereignisses definiert werden.

Syntax

Definition:

```
FUNCTION_BLOCK FB_TcSourceInfo IMPLEMENTS I_TcSourceInfo
```

Schnittstellen

Typ	Beschreibung
I_TcSourceInfo [▶ 78]	Stellt Lese-Methoden und -Eigenschaften einer Quellinformation bereit.

Methoden

Name	Definitionsort	Beschreibung
Clear [▶ 61]	Lokal	Setzt die Quellinformation zurück.
ExtendName [▶ 61]	Lokal	Fügt den übergebenen String an den Namen an.
ResetToDefault [▶ 62]	Lokal	Setzt die Eigenschaften auf Standardwerte. sName wird mit dem Symbolnamen des instanziierenden Funktionsbausteins initialisiert. nId wird mit der Objekt-ID der SPS Instanz initialisiert. Wenn die Instanz von FB_TcSourceInfo mit dem Attribut 'hide' versteckt wird, kann intern kein Symbolname für das Standardverhalten generiert werden.
EqualsTo [▶ 79]	I_TcSourceInfo [▶ 78]	Vergleicht eine Instanz mit einer anderen Instanz.

Eigenschaften

Name	Typ	Zugriff	Definitionsort	Beschreibung
guid	GUID	Get	I_TcSourceInfo [▶ 78]	Liefert die GUID der Quelleinfo zurück.
guid	GUID	SET	Lokal	Setzt die GUID als Quellinfo.
nId	UDINT	Get	I_TcSourceInfo [▶ 78]	Liefert die ID der Quellinfo.
nId	UDINT	SET	Lokal	Setzt die ID der Quellinfo.
sName	STRING(ParameterList.cSourceNameSize-1)	Get	I_TcSourceInfo [▶ 78]	Liefert den Namen der Quellinfo.
sName	STRING(ParameterList.cSourceNameSize-1)	SET	Lokal	Setzt den Namen der Quellinfo

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.20	PC oder CX (x64, x86, ARM)	Tc3_EventLogger

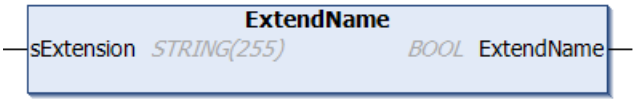
3.11.1 Clear

Diese Methode setzt die Quellinformation zurück.

Syntax

```
METHOD Clear
```

3.11.2 ExtendName



Diese Methode erweitert den Namen.

Syntax

```
METHOD ExtendName : BOOL
VAR_INPUT
    sExtension : STRING(255);
END_VAR
```

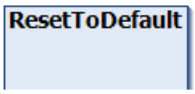
Eingänge

Name	Typ	Beschreibung
sExtension	STRING(255)	Text der rechts angehängt werden soll.

Rückgabewert

Name	Typ	Beschreibung
ExtendName	BOOL	Liefert TRUE, wenn die Verkettung erfolgreich war. Liefert FALSE, wenn die resultierende Zeichenfolge länger ist als die Ausgabe-Zeichenfolge und nicht in den gegebenen Ausgangspuffer passt. Der Speicherbedarf der resultierenden Zeichenfolge ist dann größer als der der Ausgabe-Zeichenfolge. Die Zeichenfolge wird dann abgeschnitten.

3.11.3 ResetToDefault



Diese Methode setzt die Quellinformation auf Standardwerte.

Standardwerte:

sName wird mit dem Symbolnamen des instanziierenden Funktionsbausteins initialisiert.

nId wird mit der Objekt-ID der SPS-Instanz initialisiert.

Wenn die Instanz von FB_TcSourceInfo mit dem Attribut 'hide' versteckt wird, kann intern kein Symbolname für das Standardverhalten generiert werden.

Syntax

```
METHOD ResetToDefault
```

4 Schnittstellen

4.1 I_TcArguments

Diese Schnittstelle definiert Methoden für das Argumenten-Handling.

Vererbungshierarchie

__SYSTEM.IQueryInterface

I_TcArguments

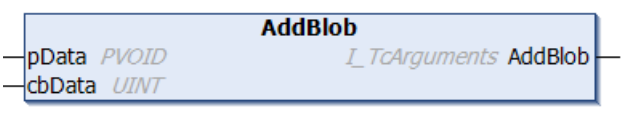
Methoden

Name	Beschreibung
AddBlob [▶ 63]	Fügt Binärdaten als Argument hinzu.
AddBool [▶ 64]	Fügt ein Argument vom Typ BOOL hinzu.
AddByte [▶ 64]	Fügt ein Argument vom Typ BYTE hinzu.
AddDint [▶ 65]	Fügt ein Argument vom Typ DINT hinzu.
AddDWord [▶ 65]	Fügt ein Argument vom Typ DWORD hinzu.
AddEventReferenceId [▶ 66]	Fügt eine Referenz zu einem anderen Ereignis als Argument hinzu.
AddEventReferenceIdGuid [▶ 66]	Fügt eine Referenz zu einem anderen Ereignis als Argument hinzu.
AddInt [▶ 67]	Fügt ein Argument vom Typ INT hinzu.
AddLInt [▶ 67]	Fügt ein Argument vom Typ LINT hinzu.
AddLReal [▶ 68]	Fügt ein Argument vom Typ LREAL hinzu.
AddReal [▶ 68]	Fügt ein Argument vom Typ REAL hinzu.
AddSInt [▶ 69]	Fügt ein Argument vom Typ SINT hinzu.
AddString [▶ 69]	Fügt ein Argument vom Typ STRING hinzu.
AddUDint [▶ 70]	Fügt ein Argument vom Typ UDINT hinzu.
AddUInt [▶ 70]	Fügt ein Argument vom Typ INT hinzu.
AddULInt [▶ 70]	Fügt ein Argument vom Typ ULINT hinzu.
AddUSInt [▶ 71]	Fügt ein Argument vom Typ USINT hinzu.
AddWord [▶ 71]	Fügt ein Argument vom Typ WORD hinzu.
AddWString [▶ 72]	Fügt ein Argument vom Typ WSTRING hinzu.
Clear [▶ 72]	Entfernt alle Argumente.

Eigenschaften

Name	Typ	Zugriff	Beschreibung
nCount	UDINT	Get	Liefert die Anzahl der übergebenen Argumente.

4.1.1 AddBlob



Diese Methode fügt Binärdaten als Argument hinzu.

Syntax

```
METHOD AddBlob : I_TcArguments
VAR_INPUT
    pData  : PVOID;
    cbData : UINT;
END_VAR
```

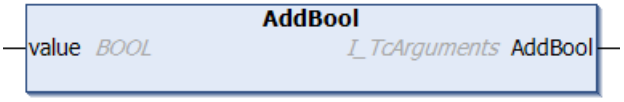
Eingänge

Name	Typ	Beschreibung
pData	PVOID	Zeiger auf das erste Byte von dem Binärdatum.
cbData	UINT	Länge des Binärdatums in Bytes.

Rückgabewert

Name	Typ	Beschreibung
AddBlob	I_TcArguments [▶ 63]	Liefert den I_TcArgument-Zeiger wieder zurück.

4.1.2 AddBool



Diese Methode fügt ein Argument vom Typ BOOL hinzu.

Syntax

```
METHOD AddBool : I_TcArguments
VAR_INPUT
    value : BOOL;
END_VAR
```

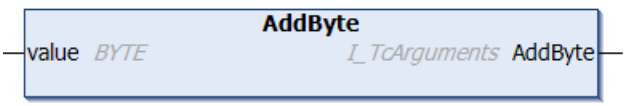
Eingänge

Name	Typ	Beschreibung
value	BOOL	Wert, der hinzugefügt werden soll.

Rückgabewert

Name	Typ	Beschreibung
AddBool	I_TcArguments [▶ 63]	Liefert den I_TcArgument-Zeiger wieder zurück.

4.1.3 AddByte



Diese Methode fügt ein Argument vom Typ BYTE hinzu.

Syntax

```
METHOD AddByte : I_TcArguments
VAR_INPUT
    value : BYTE;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
value	BYTE	Wert, der hinzugefügt werden soll.

Rückgabewert

Name	Typ	Beschreibung
AddByte	I_TcArguments [▶ 63]	Liefert den I_TcArgument-Zeiger wieder zurück.

4.1.4 AddDint

Diese Methode fügt ein Argument vom Typ DINT hinzu.

Syntax

```
METHOD AddDINT : I_TcArguments
VAR_INPUT
    value : DINT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
value	DINT	Wert, der hinzugefügt werden soll.

Rückgabewert

Name	Typ	Beschreibung
AddDINT	I_TcArguments [▶ 63]	Liefert den I_TcArgument-Zeiger wieder zurück.

4.1.5 AddDWord

Diese Methode fügt ein Argument vom Typ DWORD hinzu.

Syntax

```
METHOD AddDWord : I_TcArguments
VAR_INPUT
    value : DWORD;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
value	DWORD	Wert, der hinzugefügt werden soll.

Rückgabewert

Name	Typ	Beschreibung
AddDWord	I_TcArguments [▶ 63]	Liefert den I_TcArgument-Zeiger wieder zurück.

4.1.6 AddEventReferenceId

Diese Methode fügt eine Referenz zu einem anderen Ereignis als Argument hinzu.

Syntax

```
METHOD AddEventReferenceId : I_TcArguments
VAR_INPUT
    nEventId : UDINT;
END_VAR
```

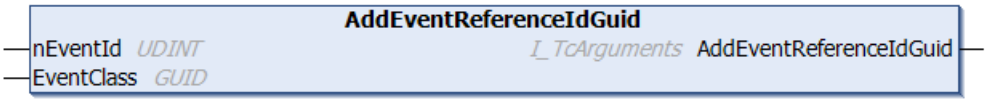
Eingänge

Name	Typ	Beschreibung
nEventId	UDINT	ID des Ereignisses.

Rückgabewert

Name	Typ	Beschreibung
AddEventReferenceId	I_TcArguments [▶ 63]	Liefert den I_TcArgument-Zeiger wieder zurück.

4.1.7 AddEventReferenceIdGuid



Diese Methode fügt eine Referenz zu einem anderen Ereignis als Argument hinzu.

Syntax

```
METHOD AddEventReferenceIdGuid : I_TcArguments
VAR_INPUT
    nEventId   : UDINT;
    EventClass : GUID;
END_VAR
```

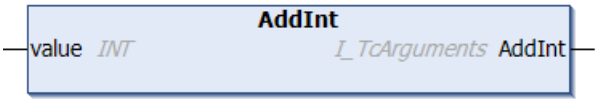
Eingänge

Name	Typ	Beschreibung
nEventId	UDINT	ID des Ereignisses.
EventClass	GUID	GUID der Ereignisklasse.

Rückgabewert

Name	Typ	Beschreibung
AddEventReferenceIdGuid	I_TcArguments [▶ 63]	Liefert den I_TcArgument-Zeiger wieder zurück.

4.1.8 AddInt



Diese Methode fügt ein Argument vom Typ INT hinzu.

Syntax

```
METHOD AddINT : I_TcArguments
VAR_INPUT
    value : INT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
value	INT	Wert, der hinzugefügt werden soll.

Rückgabewert

Name	Typ	Beschreibung
AddInt	I_TcArguments [▶ 63]	Liefert den I_TcArgument-Zeiger wieder zurück.

4.1.9 AddLInt

Diese Methode fügt ein Argument vom Typ LINT hinzu.

Syntax

```
METHOD AddLInt : I_TcArguments
VAR_INPUT
    value : LINT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
value	LINT	Wert, der hinzugefügt werden soll.

Rückgabewert

Name	Typ	Beschreibung
AddLInt	I_TcArguments [▶ 63]	Liefert den I_TcArgument-Zeiger wieder zurück.

4.1.10 AddLReal

Diese Methode fügt ein Argument vom Typ LREAL hinzu.

Syntax

```
METHOD AddLReal : I_TcArguments
VAR_INPUT
    value : LREAL;
END_VAR
```

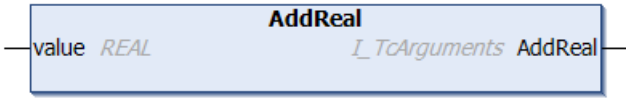
Eingänge

Name	Typ	Beschreibung
value	LREAL	Wert, der hinzugefügt werden soll.

Rückgabewert

Name	Typ	Beschreibung
AddLReal	I_TcArguments [▶ 63]	Liefert den I_TcArgument-Zeiger wieder zurück.

4.1.11 AddReal



Diese Methode fügt ein Argument vom Typ REAL hinzu.

Syntax

```
METHOD AddReal : I_TcArguments
VAR_INPUT
    value : REAL;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
value	REAL	Wert, der hinzugefügt werden soll.

Rückgabewert

Name	Typ	Beschreibung
AddReal	I_TcArguments [▶ 63]	Liefert den I_TcArgument-Zeiger wieder zurück.

4.1.12 AddSInt

Diese Methode fügt ein Argument vom Typ SINT hinzu.

Syntax

```
METHOD AddSInt : I_TcArguments
VAR_INPUT
    value : SInt;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
value	SINT	Wert, der hinzugefügt werden soll.

Rückgabewert

Name	Typ	Beschreibung
AddSInt	I_TcArguments [▶ 63]	Liefert den I_TcArgument-Zeiger wieder zurück.

4.1.13 AddString

Diese Methode fügt ein Argument vom Typ STRING hinzu.

Syntax

```
METHOD AddString : I_TcArguments
VAR_IN_OUT CONSTANT
    value : STRING;
END_VAR
```

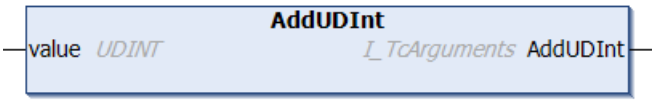
Eingänge

Name	Typ	Beschreibung
value	STRING	Wert, der hinzugefügt werden soll.

Rückgabewert

Name	Typ	Beschreibung
AddString	I_TcArguments [▶ 63]	Liefert den I_TcArgument-Zeiger wieder zurück.

4.1.14 AddUDint



Diese Methode fügt ein Argument vom Typ UDINT hinzu.

Syntax

```
METHOD AddUDInt : I_TcArguments
VAR_INPUT
    value : UDINT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
value	UDINT	Wert, der hinzugefügt werden soll.

Rückgabewert

Name	Typ	Beschreibung
AddUDInt	I_TcArguments [▶ 63]	Liefert den I_TcArgument-Zeiger wieder zurück.

4.1.15 AddUInt

Diese Methode fügt ein Argument vom Typ INT hinzu.

Syntax

```
METHOD AddUInt : I_TcArguments
VAR_INPUT
    value : UINT;
END_VAR
```

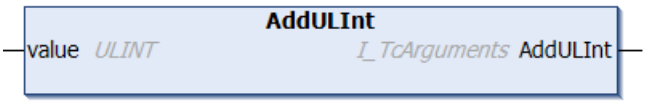
Eingänge

Name	Typ	Beschreibung
value	UINT	Wert, der hinzugefügt werden soll.

Rückgabewert

Name	Typ	Beschreibung
AddUInt	I_TcArguments [▶ 63]	Liefert den I_TcArgument-Zeiger wieder zurück.

4.1.16 AddULInt



Diese Methode fügt ein Argument vom Typ ULINT hinzu.

Syntax

```
METHOD AddULInt : I_TcArguments
VAR_INPUT
    value : ULINT;
END_VAR
```

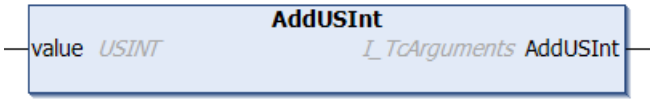
Eingänge

Name	Typ	Beschreibung
value	ULINT	Wert, der hinzugefügt werden soll.

Rückgabewert

Name	Typ	Beschreibung
AddULInt	I_TcArguments [▶ 63]	Liefert den I_TcArgument-Zeiger wieder zurück.

4.1.17 AddUSInt



Diese Methode fügt ein Argument vom Typ USINT hinzu.

Syntax

```
METHOD AddUSInt : I_TcArguments
VAR_INPUT
    value : USINT
END_VAR
```

Eingänge

Name	Typ	Beschreibung
value	USINT	Wert, der hinzugefügt werden soll.

Rückgabewert

Name	Typ	Beschreibung
AddUSInt	I_TcArguments [▶ 63]	Liefert den I_TcArgument-Zeiger wieder zurück.

4.1.18 AddWord

Diese Methode fügt ein Argument vom Typ WORD hinzu.

Syntax

```
METHOD AddWord : I_TcArguments
VAR_INPUT
    value : WORD;
END_VAR
```

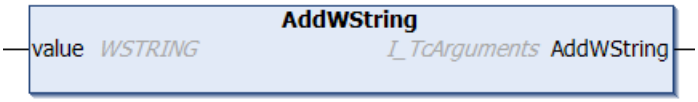
Eingänge

Name	Typ	Beschreibung
value	WORD	Wert, der hinzugefügt werden soll.

Rückgabewert

Name	Typ	Beschreibung
AddWord	I_TcArguments [▶ 63]	Liefert den I_TcArgument-Zeiger wieder zurück.

4.1.19 AddWString



Diese Methode fügt ein Argument vom Typ WSTRING hinzu.

Syntax

```
METHOD AddWString : I_TcArguments
VAR_IN_OUT CONSTANT
    value : WSTRING;
END_VAR
```

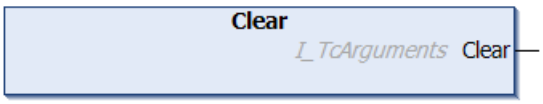
Eingänge

Name	Typ	Beschreibung
value	WSTRING	Wert, der hinzugefügt werden soll.

Rückgabewert

Name	Typ	Beschreibung
AddWString	I_TcArguments [▶ 63]	Liefert den I_TcArgument-Zeiger wieder zurück.

4.1.20 Clear



Diese Methode entfernt alle Argumente.

Syntax

```
METHOD Clear : I_TcArguments
```

Rückgabewert

Name	Typ	Beschreibung
Clear	I_TcArguments [▶ 63]	Liefert den I_TcArgument-Zeiger wieder zurück.

4.2 I_TcEventBase

In dieser Basisschnittstelle sind Methoden und Eigenschaften eines Ereignisses definiert.

Methoden

Name	Beschreibung
EqualsTo [▶ 73]	Vergleicht das Ereignis mit einer anderen Instanz.
EqualsToEventClass [▶ 74]	Vergleicht die Ereignisklasse des Ereignisses mit einer anderen Ereignisklasse.
EqualsToEventEntryEx [▶ 75]	Vergleicht die Ereignisdefinition des Ereignisses mit einer anderen Ereignisdefinition.
GetJsonAttribute [▶ 75]	Liefert das Json-Attribut.
RequestEventClassName [▶ 76]	Fragt den Namen der Ereignisklasse an.
RequestEventText [▶ 77]	Liefert den Text zum Ereignis.

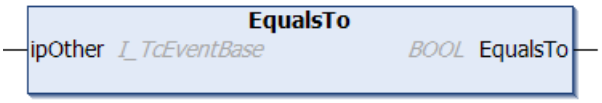
Eigenschaften

Name	Typ	Zugriff	Beschreibung
eSeverity	TcEventSeverity [▶ 80]	Get	Liefert die Severity.
EventClass	GUID	Get	Liefert die GUID der Ereignisklasse.
ipSourceInfo	I_TcSourceInfo [▶ 78]	Get	Liefert einen Zeiger auf die Quelldefinition.
nEventId	UDINT	Get	Liefert die ID des Ereignisses.
stEventEntry	TcEventEntry [▶ 80]	Get	Liefert die Ereignisdefinition.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.20	PC oder CX (x64, x86, ARM)	Tc3_EventLogger

4.2.1 EqualsTo



Diese Methode führt einen Vergleich mit einem am Eingang angegebenen anderen Ereignis aus.

Syntax

```
METHOD EqualsTo : BOOL
VAR_INPUT
    ipOther : I_TcEventBase;
END_VAR
```

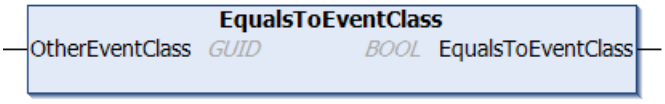
Eingänge

Name	Typ	Beschreibung
ipOther	I_TcEventBase [▶ 73]	Zu vergleichendes Ereignis

Rückgabewert

Name	Typ	Beschreibung
EqualsTo	BOOL	Liefert TRUE, wenn die Ereignisse übereinstimmen.

4.2.2 EqualsToEventClass



Diese Methode führt einen Vergleich mit einer am Eingang angegebenen anderen Ereignisklasse aus.

Syntax

```
METHOD EqualsToEventClass : BOOL
VAR_INPUT
    OtherEventClass : GUID
END_VAR
```

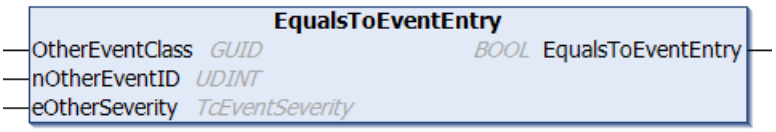
Eingänge

Name	Typ	Beschreibung
OtherEventClass	GUID	Zu vergleichende Ereignisklasse.

Rückgabewert

Name	Typ	Beschreibung
EqualsToEventClass	BOOL	Liefert TRUE, wenn die Ereignisklassen übereinstimmen.

4.2.3 EqualsToEventEntry



Diese Methode führt einen Vergleich mit einem am Eingang angegebenen anderen Ereignis aus.

Syntax

```
METHOD EqualsToEventEntry : BOOL
VAR_INPUT
    OtherEventClass : GUID;
    nOtherEventID   : UDINT;
    eOtherSeverity  : TcEventSeverity;
END_VAR
```

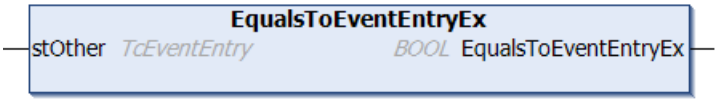
Eingänge

Name	Typ	Beschreibung
OtherEventClass	GUID	Ereignisklasse des zu vergleichenden Ereignisses.
nOtherEventID	UDINT	Event-ID des zu vergleichenden Ereignisses.
eOtherSeverity	TcEventSeverity [▶ 80]	Event-Severity des zu vergleichenden Ereignisses.

Rückgabewert

Name	Typ	Beschreibung
EqualsToEventEntry	BOOL	Liefert TRUE, wenn die Ereignisse übereinstimmen.

4.2.4 EqualsToEventEntryEx



Diese Methode führt einen Vergleich mit einem am Eingang angegebenen anderen Ereignis aus.

Syntax

```
METHOD EqualsToEventEntryEx : BOOL
VAR_INPUT
    stOther : TcEventEntry;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
stOther	TcEventEntry [▶ 80]	Zu vergleichendes Ereignis.

Rückgabewert

Name	Typ	Beschreibung
EqualsToEventEntryEx	BOOL	Liefert TRUE, wenn die Ereignisse übereinstimmen.

4.2.5 GetJsonAttribute

Diese Methode liefert das JSON-Attribut.

Syntax

```
METHOD GetJsonAttribute : HRESULT
VAR_INPUT
    sJsonAttribute : REFERENCE TO STRING;
    nJsonAttribute : UDINT;
END_VAR
```

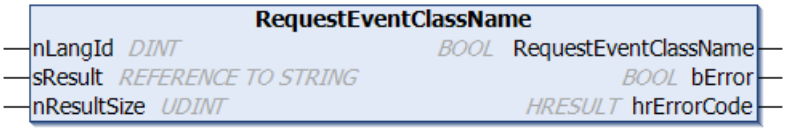
Eingänge

Name	Typ	Beschreibung
sJsonAttribute	REFERENCE TO STRING	Referenz auf eine Variable vom Typ String
nJsonAttribute	UDINT	Länge der String-Variable

Rückgabewert

Name	Typ	Beschreibung
GetJsonAttribute	HRESULT	Liefert S_OK, wenn der Methodenaufruf erfolgreich war. Liefert ERROR_BAD_LENGTH, wenn die Länge der Variable zu klein ist. Ansonsten wird HRESULT als Fehlercode zurückgegeben.

4.2.6 RequestEventClassName



Diese Methode liefert den Namen der Ereignisklasse.

Syntax

```
METHOD RequestEventClassName : BOOL
VAR_INPUT
    nLangId     : DINT;
    sResult     : REFERENCE TO STRING;
    nResultSize : UDINT;
END_VAR
VAR_OUTPUT
    bError      : BOOL;
    hrErrorCode : HRESULT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
nLangId	DINT	Spezifiziert die Sprach-ID Englisch (en-US) = 1033 Deutsch (de-DE) = 1031 …
sResult	REFERENCE TO STRING	Referenz auf eine Variable vom Typ String
nResultSize	UDINT	Größe der String-Variable in Bytes

Rückgabewert

Name	Typ	Beschreibung
RequestEventClassName	BOOL	Liefert TRUE, sobald die Abfrage beendet wurde. Wenn die asynchrone Abfrage noch aktiv ist, liefert der Rückgabewert FALSE. Die Methode muss so lange aufgerufen werden, bis der Rückgabewert TRUE ist.

Ausgänge

Name	Typ	Beschreibung
bError	BOOL	Liefert FALSE, wenn der Methodenaufruf erfolgreich war. Liefert TRUE, wenn ein Fehler aufgetreten ist.
hrErrorCode	HRESULT	Liefert S_OK, wenn der Methodenaufruf erfolgreich war. Im Falle eines Fehlers wird ein Fehlercode ausgegeben.

4.2.7 RequestEventText

Diese Methode liefert den Ereignistext.

Syntax

```
METHOD RequestEventText : BOOL
VAR_INPUT
    nLangId     : DINT;
    sResult     : REFERENCE TO STRING;
    nResultSize : UDINT;
END_VAR
VAR_OUTPUT
    bError      : BOOL;
    hrErrorCode : HRESULT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
nLangId	DINT	Spezifiziert die Sprach-ID Englisch (en-US) = 1033 Deutsch (de-DE) = 1031 …
sResult	REFERENCE TO STRING	Referenz auf eine Variable vom Typ String
nResultSize	UDINT	Größe der String-Variable in Bytes

Rückgabewert

Name	Typ	Beschreibung
RequestEventText	BOOL	Liefert TRUE, sobald die Abfrage beendet wurde. Wenn die asynchrone Abfrage noch aktiv ist, liefert der Rückgabewert FALSE. Die Methode muss so lange aufgerufen werden, bis der Rückgabewert TRUE ist.

Ausgänge

Name	Typ	Beschreibung
bError	BOOL	Liefert FALSE, wenn der Methodenaufruf erfolgreich war. Liefert TRUE, wenn ein Fehler aufgetreten ist.
hrErrorCode	HRESULT	Liefert S_OK, wenn der Methodenaufruf erfolgreich war. Im Falle eines Fehlers wird ein Fehlercode ausgegeben.

4.3 I_TcMessage

Diese Schnittstelle stellt Methoden und Eigenschaften für das Nachrichten-Handling bereit.

Vererbungshierarchie

I_TcEventBase [▶ 73]

I_TcMessage

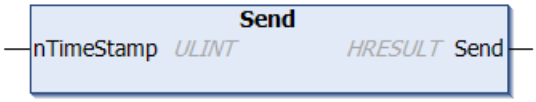
Methoden

Name	Beschreibung
Send [▶ 78]	Sendet eine Nachricht

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.20	PC oder CX (x64, x86, ARM)	Tc3_EventLogger

4.3.1 Send



Diese Methode sendet die Nachricht.

Syntax

```
METHOD Send : HRESULT
VAR_INPUT
    nTimeStamp: ULINT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
nTimeStamp	ULINT	0: Aktueller Zeitstempel wird verwendet > 0: Externer Zeitstempel in 100 Nanosekunden seit dem 1. Januar 1601 (UTC).

Rückgabewert

Name	Typ	Beschreibung
Send	FB_ HRESULT	Liefert S_OK, wenn der Methodenaufruf erfolgreich war, ansonsten ein HRESULT als Fehlercode

4.4 I_TcSourceInfo

Diese Schnittstelle definiert Eigenschaften für eine Quellinformation.

Methoden

Name	Beschreibung
EqualsTo [▶ 79]	Vergleicht eine Instanz mit Quellinformationen mit einer anderen Instanz.

Eigenschaften

Name	Typ	Zugriff	Beschreibung
guid	GUID	Get	Liefert die GUID der Quelleinfo zurück.
nId	UDINT	Get	Liefert die ID der Quelleinfo zurück.
sName	STRING(ParameterList.cSourceNameSize-1)	Get	Liefert den Namen der Quelleinfo zurück.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.20	PC oder CX (x64, x86, ARM)	Tc3_EventLogger

4.4.1 EqualsTo

Diese Methode vergleicht eine Instanz mit Quellinformationen mit einer anderen Instanz.

Syntax

```
METHOD EqualsTo : BOOL
VAR_INPUT
    ipOther : I_TcSourceInfo;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
ipOther	I_TcSourceInfo [▶ 78]	Zu vergleichende Quellinformationen

Rückgabewert

Name	Typ	Beschreibung
EqualsTo	BOOL	Liefert TRUE, wenn die Quellinformationen übereinstimmen.

5 Datentypen

5.1 TcEventEntry

Definiert ein Ereignis (Event) mittels Ereignisklasse, Ereignis-ID und Severity.

Syntax

Definition:

```
TYPE TcEventEntry :
STRUCT
    uuidEventClass : GUID;
    nEventId       : UDINT;
    eSeverity      : TcEventSeverity;
END_STRUCT
END_TYPE
```

Parameter

Name	Typ	Beschreibung
uuidEventClass	GUID	GUID der Ereignisklasse.
nEventId	UDINT	ID des Ereignisses.
eSeverity	TcEventSeverity	Event-Severity definiert den Schweregrad des Ereignisses.

5.2 TcEventSeverity

Definiert die Severity des Ereignisses.

Syntax

Definition:

```
{attribute 'qualified_only'}
TYPE TcEventSeverity : (
    Verbose  := 0,
    Info     := 1,
    Warning  := 2,
    Error    := 3,
    Critical := 4);
END_TYPE
```

Parameter

	Name	Beschreibung
4	Critical	Kritisch
3	Error	Fehler
2	Warning	Warnung
1	Info	Information
0	Verbose	Erweiterte Ausgabe

5.3 TcEventConfirmationState

Definiert den Bestätigungszustand eines Alarms.

Syntax

Definition:

```
{attribute 'qualified_only'}
TYPE TcEventConfirmationState : (
   NotSupported := 0,
   NotRequired  := 1,
   WaitForConfirmation := 2,
   Confirmed := 3,
   Reset     := 4);
END_TYPE
```

Parameter

Name	Beschreibung
Confirmed	Bestätigt
NotRequired	Bestätigung im aktuellen Zustand nicht nötig. (Alarm aktuell nicht im Zustand Raised.)
NotSupported	Wurde ohne Bestätigung initialisiert.
Reset	Initialzustand
WaitForConfirmation	Wartet auf Bestätigung.

6 Globale Listen

6.1 Global_Constants

```
VAR_GLOBAL CONSTANT
    EMPTY_EVENT_CLASS : GUID := (Data1:=16#0, Data2:=16#0, Data3:=16#0, Data4:=[16#0,16#0,16#0,16#0,
16#0,16#0,16#0,16#0]);
    EMPTY_EVENT_ID    : UDINT := 16#0;
    EMPTY_SEVERITY    : TcEventSeverity := TcEventSeverity.Verbose;
    SUCCESS_EVENT     : TcEventEntry := ( uuidEventClass := EMPTY_EVENT_CLASS, nEventID := EMPTY_EVE
NT_ID, eSeverity := EMPTY_SEVERITY );
END_VAR
```

Name	Typ	Initialwert
EMPTY_EVENT_CLASS	GUID	STRUCT(Data1:=16#0, Data2:=16#0, Data3:=16#0, Data4:=[16#0,16#0,16#0,16#0,16#0,16#0,16#0,16#0])
EMPTY_EVENT_ID	UDINT	16#0
EMPTY_SEVERITY	TcEventSeverity [▶ 80]	TcEventSeverity.Verbose
SUCCESS_EVENT	TcEventEntry [▶ 80]	STRUCT(uuidEventClass := EMPTY_EVENT_CLASS, nEventID := EMPTY_EVENT_ID, eSeverity := EMPTY_SEVERITY)

6.2 GVL

```
{attribute 'qualified_only'}
VAR_GLOBAL
    nLangId_OnlineMonitoring : DINT := 1033;
END_VAR
```

Name	Typ	Initialwert	Beschreibung
nLangId_OnlineMonitoring	DINT	1033	Sprachkennung für das Online Monitoring Englisch (en-US) = 1033 Deutsch (de-DE) = 1031 …

6.3 Parameterlist

```
{attribute 'qualified_only'}
VAR_GLOBAL CONSTANT
    cSourceNameSize : UDINT(81..10000) := 256;
END_VAR
```

Name	Typ	Initialwert	Beschreibung
cSourceNameSize	UDINT(81..10000)	256	Größe in Bytes für den Namen der Quellinformation. Empfohlen sind maximal 512 Bytes.

6.4 Global_Version

Alle Bibliotheken haben eine bestimmte Version. Diese Version ist u. a. im SPS-Bibliotheksrepository zu sehen.
Eine globale Konstante enthält die Information über die Bibliotheksversion (vom Typ ST_LibVersion):

Global_Version

```
VAR_GLOBAL CONSTANT
    stLibVersion_Tc3_EventLogger : ST_LibVersion;
END_VAR
```

Um zu sehen, ob die Version, die Sie haben auch die Version ist, die Sie brauchen, benutzen Sie die Funktion F_CmpLibVersion (definiert in der Tc2_System-Bibliothek).

7 Beispiele

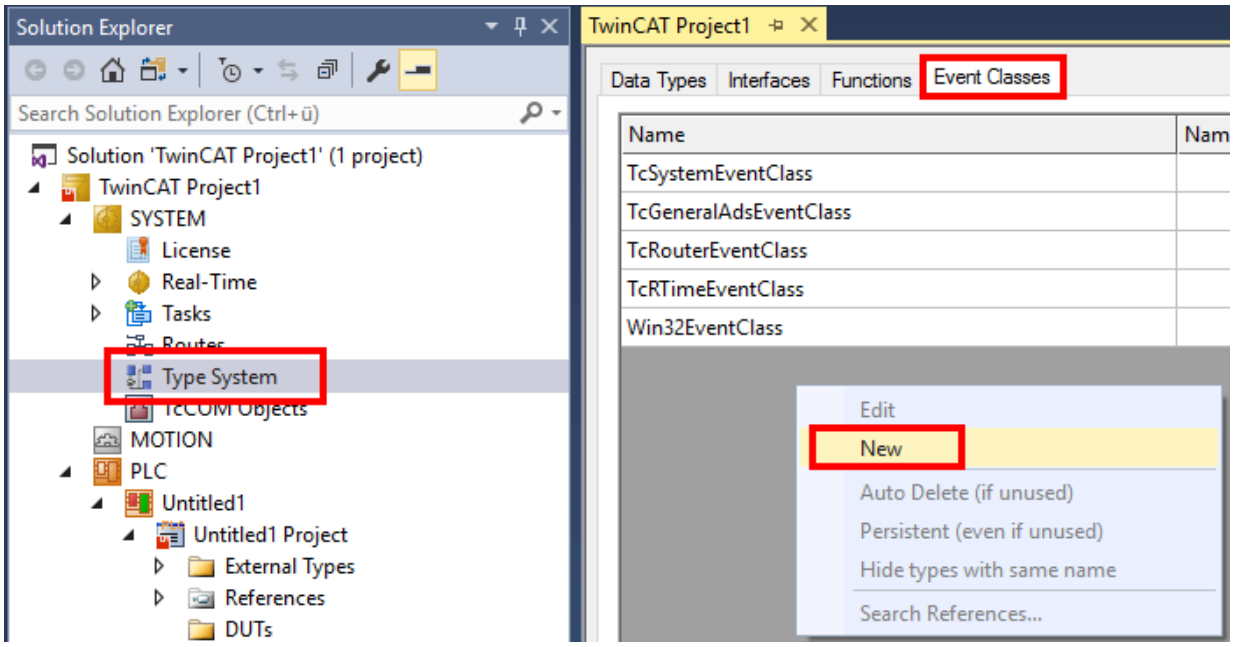
7.1 Tutorial

Dieses Tutorial verdeutlicht die Arbeitsschritte von einem leeren TwinCAT-Projekt bis hin zu einer abgesendeten Meldung. Es zeigt die im Abschnitt Technische Einführung beschriebenen Eigenschaften des TwinCAT 3 EventLoggers anschaulich im Arbeitsablauf.

Ereignisklasse im Typsystem von TwinCAT anlegen

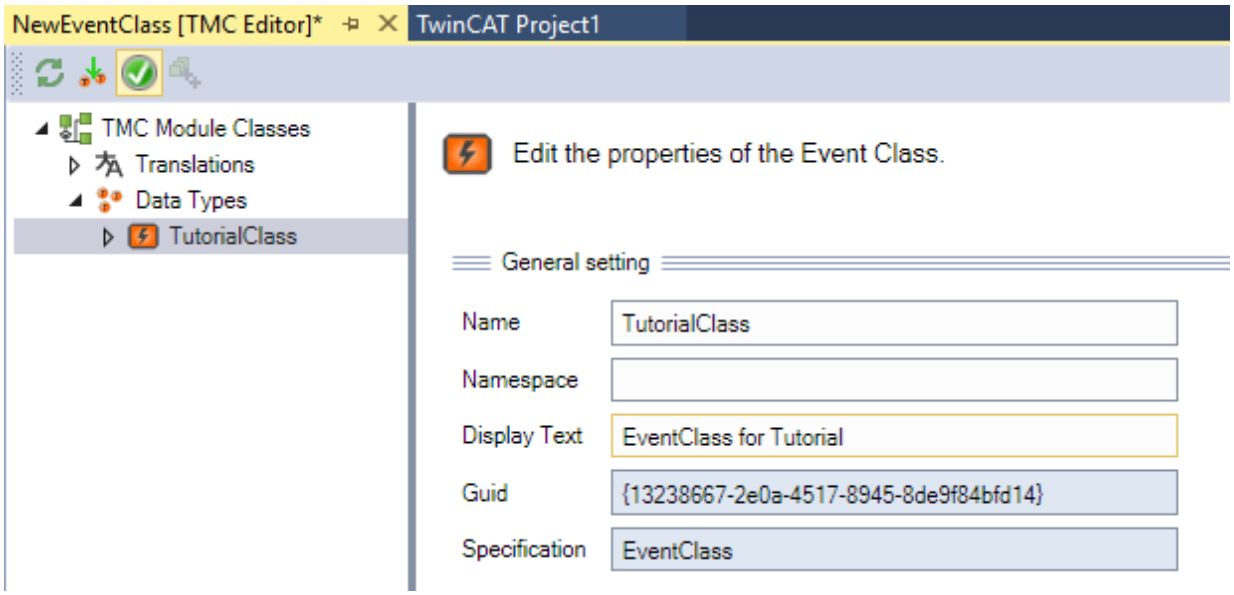
✓ Ein Standard-TwinCAT-SPS-Projekt existiert.

1. Klicken Sie im SYSTEM-Teilbaum doppelt auf **Type System** und wählen Sie in dem sich öffnenden Editor die Registerkarte **Event Classes**. Öffnen Sie das Kontextmenü und wählen Sie den Befehl **New**.

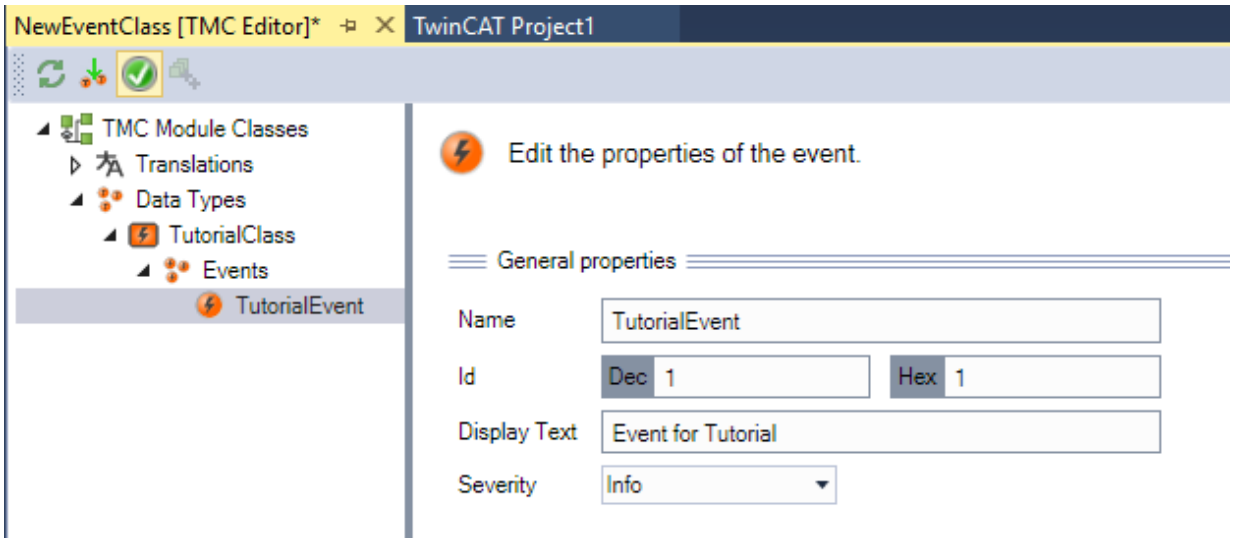


⇨ Der TMC Editor öffnet sich.

2. Geben Sie der Ereignisklasse einen Namen und geben Sie einen Display-Text an.



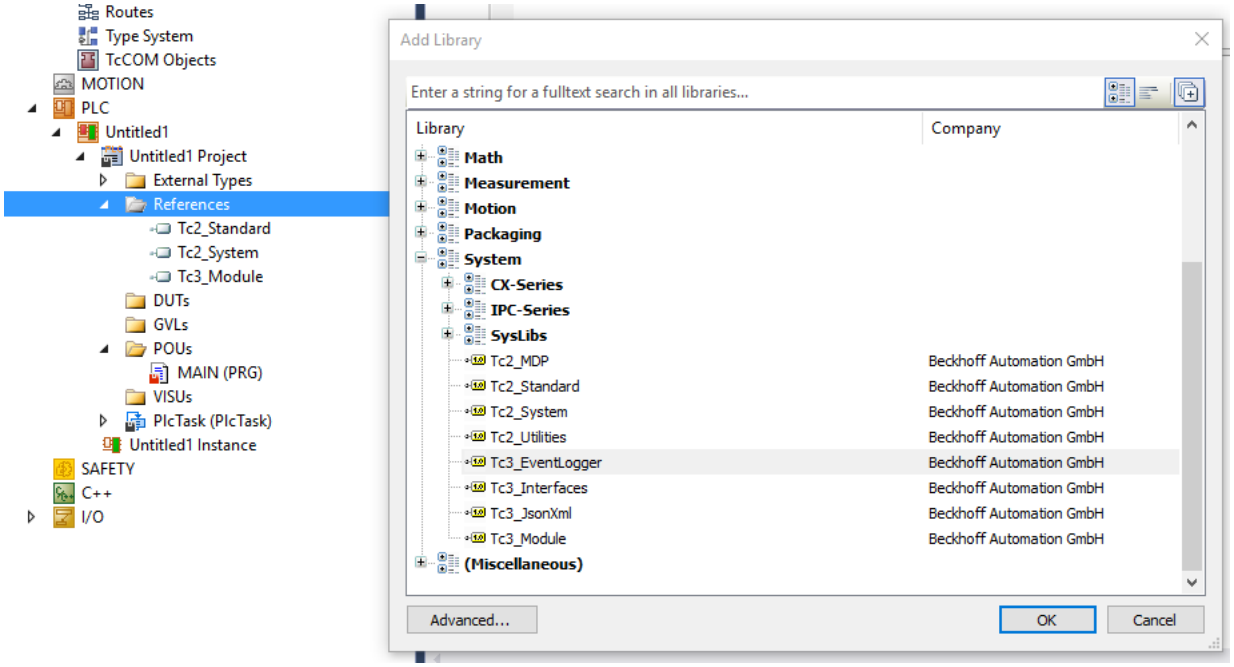
3. Unterhalb der der Ereignisklasse ist bereits ein Ereignis angelegt. Geben Sie dem Ereignis einen Namen und geben Sie einen Dispaly-Text und die Severity an.



4. Speichern und ggf. schließen Sie die Ereignisklasse.
   ⇨ Der Quellcode wird in der SPS bereitgestellt und ist unter dem Symbol TC_EVENTS erreichbar.

Bibliothek TC3_EventLogger hinzufügen

5. Wählen Sie im Kontextmenü des Objekts **References** den Befehl **Add Library**.
   ⇨ Der Dialog **Add Library** öffnet sich.
6. Wählen Sie die Bibliothek aus und bestätigen Sie den Dialog.



⇨ Die Bibliothek wird dem SPS-Projekt hinzugefügt.

SPS-Programm erstellen

1. Öffnen Sie mit einem Doppelklick das MAIN-Programm des SPS-Projekts im Editor.
2. Deklarieren und initialisieren Sie die Variablen bInit und bSend und deklarieren Sie eine Instanz des Funktionsbausteins FB_TcMessage:

```
PROGRAM MAIN
VAR
    bInit : BOOL := TRUE;
    bSend : BOOL := TRUE;
```

```
    fbMsg : FB_TcMessage;
END_VAR
```

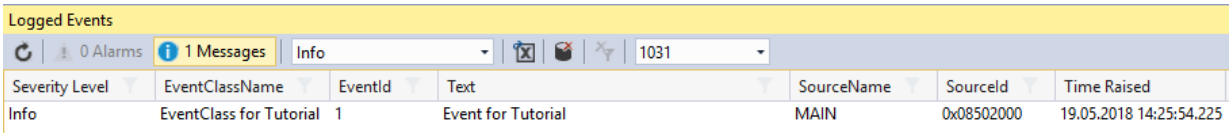
3. Implementieren Sie den Sendevorgang wie im Code dargestellt. Die Nachricht wird einmalig mittels der CreateEx-Methode initialisiert. Da die Initialisierung dynamische Ressourcen benötigt, sollte sie nicht zyklisch erfolgen. Die initialisierte Nachricht wird anschließend mit der Send-Methode gesendet.

```
IF bInit THEN
    bInit := FALSE;
    fbMsg.CreateEx(TC_EVENTS.TutorialClass.TutorialEvent, 0);
END_IF

IF bSend THEN
    bSend := FALSE;
    fbMsg.Send(0);
END_IF
```

4. Erstellen Sie das SPS-Projekt und starten Sie die SPS.

⇨ Das Ergebnis wird im Fenster LoggedEvents des TwinCAT 3 Engineerings anzeigt.



7.2 Beispiel ResultMessage

Dieses Beispiel zeigt die Verwendung des TwinCAT 3 EventLoggers mit Funktionsbausteinen. Es demonstriert zum einen, wie ein Ausgang an einem Funktionsbaustein dazu verwendet werden kann, die Event-Informationen als erweiterte Rückgabe zu nutzen. Zum anderen demonstriert es, wie eine Parametrisierung vorgenommen werden kann, um eine Ausgabe der Meldungen über den TwinCAT 3 EventLogger nur in bestimmten Fällen durchzuführen.

Download: https://infosys.beckhoff.com/content/1031/tcplclib_tc3_eventlogger/Resources/5288319115.zip

Das Bespiel besteht aus zwei Funktionsbausteinen:

- FB_MathCalculation: Dieser Funktionsbaustein bietet zwei Methoden und zwei Properties an, die Meldungen immer am Ausgang ipResultMessage ausgeben und zusätzlich über den EventLogger absenden, wenn ein Tracelevel überschritten ist.
  - Methode Addition(): Addiert zwei Zahlen und sendet bei einem Überlauf eine Nachricht
  - Methode Divison(): Dividiert zwei Zahlen nach Prüfung. Sendet eine Nachricht, wenn eine Division durch 0 erfolgt.
  - Property bTraceLevelDefault: Gibt an, ob das Tracelevel lokal am Funktionsbaustein beachtet werden soll, oder ob ein Library Tracelevel verwendet werden soll, welcher im Beispiel in der GVL vorhanden ist.
  - Property eTraceLevel: Die Methoden senden die Nachricht nur über den EventLogger ab, wenn die Severity größer oder gleich diesem Property ist.
- FB_Control: Dieser Funktionsbaustein zeigt die Verwendung des FB_MathCalculation-Bausteins innerhalb eines anderen Bausteins. Dabei nutzt die Execute-Methode des FB_Control die FB_MathCalculation.Divison() und behandelt die Nachricht als Fehlercode selbst weiter.

7.3 Beispiel Listener

Dieses Beispiel zeigt die Verwendung des TwinCAT 3 EventLoggers in Bezug auf Nachrichten und Alarme. Gleichzeitig wird das Empfangen von Nachrichten in einem zweiten Projekt gezeigt.

Download: https://infosys.beckhoff.com/content/1031/tcplclib_tc3_eventlogger/Resources/5288316939.zip

Publisher-Projekt

Im Publisher-Projekt werden einfache BOOL-Variablen als Trigger verwendet:

- bSendMessage, um eine Nachricht abzusetzen.
- bRaiseAlarm, um einen Alarm zu setzen.
- bClearAlarm, um einen Alarm zurückzunehmen.
- bConfirmAlarm, um einen Alarm zu quittieren.

Zusätzlich gibt es die Möglichkeit, das JSON-Attribut zu setzen, um dieses bei beiden Nachrichten mitzusenden.

Listener-Projekt

Im Listener-Projekt ist ein Funktionsbaustein FB_Listener enthalten, der den in der Tc3_EventLogger enthaltenen Baustein FB_ListenerBase erweitert. Der Baustein implementiert hierbei die Funktionen zum Empfang der Nachrichten:

- OnMessageSent: Wenn eine Nachricht versendet wurde, wird der EventLogger diese Methode als Callback aufrufen. Die Methode zählt die Anzahl der Nachrichten mit.
- OnAlarmRaised/OnAlarmCleared/OnAlarmConfirmed: Wenn der Alarm den Zustand ändert, wird der EventLogger diese Methode als Callback aufrufen. Die Methoden zählen jeweils die Anzahl der Zustandsänderungen mit.
- Um den Empfang der Nachrichten zu initiieren, ist eine Execute-Methode an dem Baustein implementiert.
- Der Text der letzten empfangenen Nachricht kann abgeholt werden.
- Der Funktionsbaustein FB_ListenerTest nutzt den FB_Listener. Hierbei registriert er einmalig die zu empfangende Ereignisklasse. Eine weitere existierende Ereignisklasse wird nicht empfangen, wodurch die Filterfunktionalität demonstriert wird.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.17	PC oder CX (x64, x86, ARM)	Tc3_EventLogger (>= v3.1.27.0)

7.4 Beispiel Filter

Dieses Beispiel zeigt die Verwendung des TwinCAT 3 EventLoggers in Bezug auf das Empfangen von Nachrichten. Hierbei wird ein Fokus auf die Filterfunktionen gesetzt, um zielgerichtet die richtigen Nachrichten zu verarbeiten.

Download: https://infosys.beckhoff.com/content/1031/tcplclib_tc3_eventlogger/Resources/10400437387.zip

Das Beispiel besteht aus vier Komponenten:

- Es werden eine Reihe von unterschiedlichen Nachrichten abgesendet, wodurch die Selektion der Nachrichten in unterschiedlichen Filtern demonstriert wird.
- Eine Komponente zeigt, wie aus dem Cache Nachrichten verworfen werden können, welche über einen Filter spezifiziert werden.
- Eine andere Komponente zeigt den Export von im Cache hinterlegten Nachrichten in eine CSV-Datei. Auch hierbei wird über die Filter programmiert, welche Nachrichten ausgewählt werden sollen.
- Eine weitere Komponente zeigt das allgemeine Empfangen von in der Echtzeit gesendeten Nachrichten sowie das Empfangen von EtherCAT Emergency Nachrichten, welche empfangen werden sollen.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.17	PC oder CX (x64, x86, ARM)	Tc3_EventLogger (>= v3.1.27.0)

Mehr Informationen:
www.beckhoff.de/te1000

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