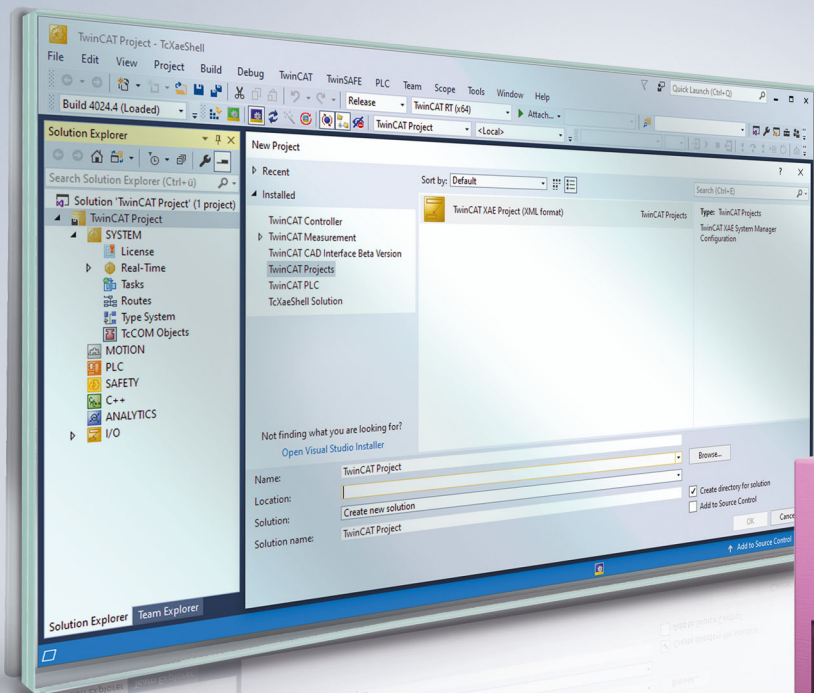


BECKHOFF New Automation Technology

Handbuch | DE

TF6510

TwinCAT 3 | IEC 61850



Inhaltsverzeichnis

1	Vorwort.....	11
1.1	Hinweise zur Dokumentation	11
1.2	Zu Ihrer Sicherheit.....	12
1.3	Hinweise zur Informationssicherheit	13
2	Übersicht.....	14
3	Installation	15
3.1	Systemvoraussetzungen	15
3.2	Installation	15
3.3	Installation Windows CE	18
3.4	Lizenzierung	20
4	Technische Einführung	23
4.1	Normübersicht.....	23
4.2	Datenmodell	24
4.3	Unterstützte Services	26
4.4	Allgemeine Client - Projektstruktur	27
4.5	Allgemeine Server - Projektstruktur	32
4.6	RT - Ethernet Adapter Konfiguration	37
4.7	TCP Keep-Alive Messages	39
4.8	Firewall Einstellungen	41
5	Konfiguration.....	43
5.1	Grundlagen und Einstieg.....	43
5.2	Menüleiste.....	50
5.2.1	File	50
5.2.2	View	51
5.2.3	Tools	52
5.2.4	Help.....	52
5.3	New Project-Wizard	53
5.4	Konfiguration der Datenobjekte.....	60
5.5	DataSets	65
5.6	Reporting.....	70
5.7	GOOSE	72
5.8	Automatische Code-Erzeugung	74
5.8.1	Codeversion 2.0.....	79
5.9	Benutzerdefinierte Logical Node Groups	91
5.10	Benutzerdefinierte Logical Nodes	91
5.11	Benutzerdefinierte Common Data Classes	93
6	SPS-API.....	96
6.1	Bibliotheken.....	96
6.2	Namensräume.....	96
6.2.1	TwinCAT TypeSystem Namensraum.....	96
6.2.2	Tc3_Acsi Namensraum.....	96
6.2.3	Tc3_Ber Namensraum	96
6.2.4	Tc3_Collections Namensraum	96

6.2.5	Tc3_Gse Namensraum	97
6.2.6	Tc3_iec61850_8_1 Namensraum	97
6.2.7	Tc3_iec61850 Namensraum	97
6.2.8	Tc3_Mms Namensraum	97
6.2.9	Tc3_Rfc1006 Namensraum	97
6.2.10	Tc3_Sockets Namensraum	97
6.2.11	Tc3_Tpkt Namensraum	97
6.2.12	Tc3_Ulosi Namensraum	97
6.2.13	Tc2_Tcplp Namensraum	97
6.2.14	Tc2_System Namensraum	97
6.3	Funktionsbausteine	98
6.3.1	Datenmodell	98
6.3.2	Protokolleinstellungen	166
6.3.3	Kommunikation	220
6.3.4	FB_AsyncServiceResultClass	244
6.3.5	FB_ServiceErrorClass	245
6.4	Funktionen	246
6.4.1	Accuracy_To_UtcTimeQualityAccuracy	246
6.4.2	AcsiAnalogueValue	248
6.4.3	AcsiCalendarTime	248
6.4.4	AcsiCell	249
6.4.5	AcsiLogOptionalFields	250
6.4.6	AcsiOptionalFields	250
6.4.7	AcsiOriginator	251
6.4.8	AcsiPhyComAddr	252
6.4.9	AcsiPoint	252
6.4.10	AcsiPulseConfig	253
6.4.11	AcsiQuality	254
6.4.12	AcsiQuality_To_String	255
6.4.13	AcsiQuality_To_WORD	255
6.4.14	AcsiQualitySource_To_Source	256
6.4.15	AcsiQualityValidity_To_Validity	257
6.4.16	AcsiRangeConfig	258
6.4.17	AcsiReasonCode	259
6.4.18	AcsiScaledValueConfig	260
6.4.19	AcsiSvOptionalFields	260
6.4.20	AcsiTriggerConditions	261
6.4.21	AcsiUnit	262
6.4.22	AcsiValWithTrans	262
6.4.23	AcsiVector	263
6.4.24	AnyBuffer_To_OctetString	263
6.4.25	BinaryTime	265
6.4.26	BinaryTime_To_SystemTime	265
6.4.27	BinaryTime6_To_LWORD	266
6.4.28	BinaryTime6_To_String	267
6.4.29	BinaryTime6Day_To_Date	268

6.4.30	BYTE_To_UtcTimeQuality	269
6.4.31	CltsEveryBool.....	270
6.4.32	CltsSomeBool	272
6.4.33	DATE_and_TOD_To_DT	273
6.4.34	Date_To_BinaryTime6Day.....	274
6.4.35	FillOctetString.....	275
6.4.36	INT24	275
6.4.37	INT128	276
6.4.38	LTIME_To_UtcTimeFractionOfSecond	277
6.4.39	LWORD_To_BinaryTime6	278
6.4.40	LWORD_TO_MACADDR.....	279
6.4.41	LWORD_TO_OCTET6.....	280
6.4.42	LWORD_TO_OCTET8.....	280
6.4.43	LWORD_To_UtcTime	281
6.4.44	OctetString_To_AnyBuffer	282
6.4.45	SizeOfOctetString	283
6.4.46	Source_To_AcsiQualitySource	283
6.4.47	String_To_AcsiQuality.....	284
6.4.48	String_To_BinaryTime6	285
6.4.49	String_To_UtcTime	286
6.4.50	String_To_UtcTimeQuality	288
6.4.51	String_To_UtcTimeStamp.....	289
6.4.52	SystemTime_To_BinaryTime.....	290
6.4.53	SystemTime_To_UtcTime.....	291
6.4.54	UINT24.....	292
6.4.55	UINT128.....	293
6.4.56	UtcTime.....	293
6.4.57	UtcTime_To_LWORD	294
6.4.58	UtcTime_To_String	295
6.4.59	UtcTime_To_SystemTime.....	296
6.4.60	UtcTimeFractionOfSecond_To_LTIME	298
6.4.61	UtcTimelsNull.....	299
6.4.62	UtcTimeQuality.....	299
6.4.63	UtcTimeQuality_To_BYTE	300
6.4.64	UtcTimeQuality_To_String	301
6.4.65	UtcTimeQualityAccuracy_To_Accuracy.....	303
6.4.66	UtcTimeStamp_To_String.....	304
6.4.67	Validity_To_AcsiQualityValidity.....	305
6.4.68	WORD_To_AcsiQuality.....	306
6.4.69	ZeroOctetString.....	307
6.5	Schnittstellen.....	308
6.5.1	Datenmodell	308
6.5.2	Protokolleinstellungen	322
6.5.3	Kommunikation	337
6.5.4	I_AsyncServiceResultClass	387
6.5.5	I_ServiceErrorClass	392

6.5.6	I_BaseStreamClass	395
6.6	Datentypen	395
6.6.1	E_AcsiAccessPermission	395
6.6.2	E_AcsiAdjSt	396
6.6.3	E_AcsiAlmLev	396
6.6.4	E_AcsiAnalogueValuePresCond	397
6.6.5	E_AcsiAngRef	397
6.6.6	E_AcsiApplAddCause	398
6.6.7	E_AcsiApplError	399
6.6.8	E_AcsiAutoRecSt	400
6.6.9	E_AcsiBeh	400
6.6.10	E_AcsiCBOpCap	401
6.6.11	E_AcsiCheck	401
6.6.12	E_AcsiClcIntvTyp	402
6.6.13	E_AcsiClcMod	402
6.6.14	E_AcsiClcMth	403
6.6.15	E_AcsiClcRfTyp	403
6.6.16	E_AcsiClcTotVA	404
6.6.17	E_AcsiCmdQual	404
6.6.18	E_AcsiCtlModels	405
6.6.19	E_AcsiCtlReport	405
6.6.20	E_AcsiCtlService	405
6.6.21	E_AcsiCycTrMod	406
6.6.22	E_AcsiDbpos	406
6.6.23	E_AcsiDir	407
6.6.24	E_AcsiDirMod	407
6.6.25	E_AcsiEEHealth	408
6.6.26	E_AcsiFailMod	408
6.6.27	E_AcsiFanCtl	408
6.6.28	E_AcsiFanCtlGen	409
6.6.29	E_AcsiFc	409
6.6.30	E_AcsiFilTyp	410
6.6.31	E_AcsiFltLoop	411
6.6.32	E_AcsiGnSt	411
6.6.33	E_AcsiHealth	412
6.6.34	E_AcsiHvRef	412
6.6.35	E_AcsiIntrDetMth	412
6.6.36	E_AcsiLevMod	413
6.6.37	E_AcsiLivDeaMod	413
6.6.38	E_AcsiLogicalNodeClass	414
6.6.39	E_AcsiMechHealth	415
6.6.40	E_AcsiMod	415
6.6.41	E_AcsiMonth	415
6.6.42	E_AcsiMultiplier	416
6.6.43	E_AcsiOccPer	417
6.6.44	E_AcsiOccType	417

6.6.45	E_AcsiOpModRect.....	418
6.6.46	E_AcsiOpModSyn.....	418
6.6.47	E_AcsiOrCategory.....	419
6.6.48	E_AcsiParColMod.....	419
6.6.49	E_AcsiParMod.....	420
6.6.50	E_AcsiParTraMod.....	420
6.6.51	E_AcsiPFSign.....	421
6.6.52	E_AcsiPhsRef.....	421
6.6.53	E_AcsiPhyHealth.....	422
6.6.54	E_AcsiPIDAlg.....	422
6.6.55	E_AcsiPmpCtl.....	422
6.6.56	E_AcsiPmpCtlGen.....	423
6.6.57	E_AcsiPolQty.....	423
6.6.58	E_AcsiPOWCap.....	424
6.6.59	E_AcsiQualitySource.....	424
6.6.60	E_AcsiQualityValidity.....	425
6.6.61	E_AcsiRange.....	425
6.6.62	E_AcsiRcdMod.....	425
6.6.63	E_AcsiReTrMod.....	426
6.6.64	E_AcsiRotDir.....	426
6.6.65	E_AcsiRstMod.....	427
6.6.66	E_AcsiSboClasses.....	427
6.6.67	E_AcsiSeqT.....	428
6.6.68	E_AcsiServerDirectoryClass.....	428
6.6.69	E_AcsiServiceError.....	428
6.6.70	E_AcsiServiceType.....	429
6.6.71	E_AcsiSetCharact.....	430
6.6.72	E_AcsiSev.....	431
6.6.73	E_AcsiShOpCap.....	431
6.6.74	E_AcsiSIUnit.....	432
6.6.75	E_AcsiSptEndSt.....	434
6.6.76	E_AcsiStClcTun.....	435
6.6.77	E_AcsiStrWeekDay.....	436
6.6.78	E_AcsiSwOpCap.....	436
6.6.79	E_AcsiSwTyp.....	437
6.6.80	E_AcsiTcmd.....	437
6.6.81	E_AcsiTmSyn.....	438
6.6.82	E_AcsiTnkTyp.....	438
6.6.83	E_AcsiTpcRxMod.....	438
6.6.84	E_AcsiTpcTxMod.....	439
6.6.85	E_AcsiTrBeh.....	439
6.6.86	E_AcsiTrgMod.....	440
6.6.87	E_AcsiTrMod.....	440
6.6.88	E_AcsiTypRsCrv.....	440
6.6.89	E_AcsiUnbDetMth.....	441
6.6.90	E_AcsiUnBlkMod.....	441

6.6.91	E_AcsiVlanPriority	442
6.6.92	E_AcsiWeekDay	442
6.6.93	E_AcsiWeiMod	443
6.6.94	E_AsyncEnvironmentState	443
6.6.95	E_AsyncServiceState	444
6.6.96	E_EventCompletion	444
6.6.97	E_GseDispatchMode	445
6.6.98	E_GseDispatchMode	445
6.6.99	E_GseGoCBPublisherState	446
6.6.100	E_GseGoCBSubscriberState	446
6.6.101	E_GseLinkStatus	446
6.6.102	E_GseRetransmissionStrategy	447
6.6.103	E_MmsVmdLogicalStatus	447
6.6.104	E_MmsVmdPhysicalStatus	447
6.6.105	E_Rfc1006TpduSize	448
6.6.106	E_ScsmEdition	448
6.6.107	E_TraceLevel	449
6.6.108	E_UtcTimeAccuracy	449
6.6.109	Externe Datentypen	451
6.6.110	ST_AcsiAnalogueValue	455
6.6.111	ST_AcsiCalendarTime	455
6.6.112	ST_AcsiCell	455
6.6.113	ST_AcsiDataSetClassInfo	456
6.6.114	ST_AcsiFunctionalConstraints	456
6.6.115	ST_AcsiIEDGroupClassInfo	457
6.6.116	ST_AcsiIntelligentElectronicDeviceClassInfo	458
6.6.117	ST_AcsiLogicalDeviceClassInfo	458
6.6.118	ST_AcsiLogicalNodeClassInfo	458
6.6.119	ST_AcsiLogOptionalFields	459
6.6.120	ST_AcsiOptionalFields	459
6.6.121	ST_AcsiOriginator	460
6.6.122	ST_AcsiPhyComAddr	460
6.6.123	ST_AcsiPoint	461
6.6.124	ST_AcsiPresCond	461
6.6.125	ST_AcsiPulseConfig	462
6.6.126	ST_AcsiQuality	463
6.6.127	ST_AcsiRangeConfig	464
6.6.128	ST_AcsiReasonCode	465
6.6.129	ST_AcsiScaledValueConfig	465
6.6.130	ST_AcsiSvOptionalFields	466
6.6.131	ST_AcsiTriggerConditions	466
6.6.132	ST_AcsiUnit	467
6.6.133	ST_AcsiValWithTrans	467
6.6.134	ST_AcsiVector	468
6.6.135	ST_GseAdapterInfo	468
6.6.136	ST_GseGoCBPublisherTx	468

6.6.137	ST_GseGoCBSsubscriberRx.....	470
6.6.138	ST_ScsmAdditionalCauseDiagnosticEventInfo	471
6.6.139	ST_ScsmBrCBJobInfo	471
6.6.140	ST_ScsmBrCBValuesSet.....	471
6.6.141	ST_ScsmCommandTerminationEventInfo.....	472
6.6.142	ST_ScsmConfirmedEventInfo.....	472
6.6.143	ST_ScsmGoCBJobInfo	473
6.6.144	ST_ScsmGoCBValuesSet	473
6.6.145	ST_ScsmReportEventInfo.....	474
6.6.146	ST_ScsmUrCBJobInfo	475
6.6.147	ST_ScsmUrCBValuesSet	475
6.6.148	ST_ServiceErrorClass.....	475
6.6.149	T_AcsiObjectName	475
6.6.150	T_AcsiObjectReference	476
6.6.151	T_AcsiTag	476
6.6.152	T_AcsiVisString129.....	476
6.6.153	T_BinaryTime	476
6.6.154	T_INT24	477
6.6.155	T_INT128	477
6.6.156	T_MmsInteger8.....	477
6.6.157	T_MmsInteger16.....	477
6.6.158	T_MmsInteger32.....	478
6.6.159	T_MmsString.....	478
6.6.160	T_MmsVisibleString	478
6.6.161	T_OCTET6.....	478
6.6.162	T_OCTET8.....	478
6.6.163	T_OCTET16.....	478
6.6.164	T_OCTET32.....	479
6.6.165	T_OCTET64.....	479
6.6.166	T_UINT24.....	479
6.6.167	T_UINT128.....	479
6.6.168	T_UtcTime.....	479
6.6.169	T_UtcTimeQuality	480
6.6.170	U_AcsiAnalogueCtlVal	481
6.6.171	U_AcsiCtlVal	481
6.6.172	U_AcsiEnumeratedCtlVal.....	482
6.7	Parameterlisten	483
6.7.1	Param_Acsi.....	483
6.7.2	Param_Gse	489
6.7.3	Param_Sockets.....	489
6.8	Globale Variablen.....	489
6.8.1	TC_EVENTS Konstanten.....	489
6.8.2	TC_EVENT_CLASSES Konstanten.....	583
6.8.3	GVL_Acsi Konstanten	586
7	Beispiele	591
7.1	Allgemeine Client - Projektstruktur	591

7.2	Client - Basis Beispielprojekt.....	596
7.3	Client - Read Data (GetAllDataValues, GetDataValues)	597
7.4	Client - Write Data (SetAllDataValues, SetDataValues)	599
7.5	Client - Report Control Blocks (Unbuffered, Buffered)	601
7.6	Client - Direct Control with normal security	612
7.7	Client - SBO Control with normal security	613
7.8	Client - Direct Control with enhanced security	614
7.9	Client - SBO Control with enhanced security	616
7.10	Client - Read/Write DataSet Values (GetDataSetValues, SetDataSetValues)	617
7.11	Client - GOOSE Subscriber (GetGoCBDDataValues, SetGoCBDDataValues)	621
7.12	Client - GOOSE Subscriber (ohne Client-Server-Kommunikation)	622
7.13	Client - User defined timestamp clock source	623
7.14	Client - Dynamic created DataSets (CreateDataSet, DeleteDataSet)	625
7.15	Allgemeine Server - Projektstruktur	628
7.16	Server - Basis Beispielprojekt	632
7.17	Server – GOOSE Publisher (GetGoCBDDataValues, SetGoCBDDataValues)	632
7.18	Server - GOOSE Publisher (ohne Client-Server-Kommunikation)	633
7.19	Server - Direct Control with normal security	635
7.20	Server - Direct Control with enhanced security	639
7.21	Server - SBO Control with normal security	644
7.22	Server - SBO Control with enhanced security	650
7.23	Server - User defined timestamp clock source	656
7.24	Server - Unbuffered Reporting	658
7.25	Server - Ereignisse bei Datenzugriff	660
7.26	Server - Buffered Reporting	662
7.27	Mehrere GOOSE Publisher/Subscriber an einem Netzwerkadapter (ohne Client-Server-Kommunikation)	666
7.28	TLS – Sichere Client-Server-Kommunikation mit Hilfe von Zertifikaten oder PSK (Pre-Shared Key)	674
8	Anhang	681
8.1	Rückgabewerte	681
8.2	Binary-Time-LWORD-Formatspezifikation	681
8.3	Binary-Time-String-Formatspezifikation	681
8.4	UTC-Time-LWORD-Formatspezifikation	682
8.5	UTC-Time-Quality-BYTE-Formatspezifikation	684
8.6	UTC-Time-Quality-String-Formatspezifikation	685
8.7	UTC-Time-String-Formatspezifikation	686
8.8	Quality-String-Formatspezifikation	687
8.9	Quality-WORD-Formatspezifikation	688
8.10	FAQ - Häufig gestellte Fragen und Antworten	689

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

Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

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

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

1.2 Zu Ihrer Sicherheit

Sicherheitsbestimmungen

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

Haftungsausschluss

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

Qualifikation des Personals

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

Signalwörter

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

Warnungen vor Personenschäden

GEFAHR

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

WARNUNG

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

VORSICHT

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

Warnung vor Umwelt- oder Sachschäden

HINWEIS

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

Information zum Umgang mit dem Produkt



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

1.3 Hinweise zur Informationssicherheit

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

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

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

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

2 Übersicht

Die IEC 61850 ist ein Kommunikationsprotokoll für den Datenaustausch in elektrischen Schaltanlagen. Das TwinCAT-Produkt zur IEC 61850 bietet diese Art der Kommunikation direkt aus der TwinCAT-Steuerung in Form einer SPS-Programmierschnittstelle. Der gesamte Kommunikationsstack ist in der SPS abgebildet. Das Produkt kann auch auf die auf der IEC 61850 aufbauenden Protokolle angewendet werden. Dies gilt insbesondere im Bereich der Windkraftanlagen für die Datenmodelle der IEC 61400-25. Grundsätzlich gibt es neben der SPS-Programmierschnittstelle noch einen Konfigurator für das Datenmodell. Dieser kann die icd-Dateien der IEC 61850 einlesen und auch generieren. Ebenso kann er den notwendigen SPS-Code für die Abbildung des Datenmodells automatisch erzeugen, wodurch sich der Anwender auf die eigentliche Applikation konzentrieren kann. Neben der Client/Server Kommunikation unterstützt TwinCAT auch das direkt auf dem Ethernet Layer aufsetzende GOOSE Protokoll.

Produktkomponenten

Das Produkt TF6510 IEC 61850 besteht aus den folgenden Komponenten:

- **SPS-Bibliotheken:**
 - Tc3_IEC61850.compiled-library,
 - Tc3_IEC61850_8_1.compiled-library,
 - Tc3_Gse.compiled-library,
 - Tc3_Mms.compiled-library,
 - Tc3_Acsi.compiled-library,
 - Tc3_Ber.compiled-library,
 - Tc3_Rfc1006.compiled-library,
 - Tc3_Collections.compiled-library,
 - Tc3_Tpkt.compiled-library,
 - Tc3_Ulosi.compiled-library,
- **Andere Produkte:** TF6310 TCP/IP Server
- **TwinCAT Telecontrol Configurator:** Konfigurator für das IEC 61850/ IEC 61400-25 Datenmodell

3 Installation

3.1 Systemvoraussetzungen

Entwicklungsumgebung

Eine reine Entwicklungsumgebung beschreibt einen Computer, auf dem SPS-Programme entwickelt, aber nicht ausgeführt werden. Auf einem Entwicklungscomputer muss Folgendes installiert sein:

- TwinCAT 3.1.4024.0 XAE (Engineering) oder höher
- TwinCAT 3 Function TF6510
- Für die Entwicklungsumgebung kann (auch mehrfach) eine 7-Tage Testlizenz verwendet werden (siehe [Lizenzierung \[► 20\]](#)).

Laufzeitumgebung

Eine Laufzeitumgebung beschreibt einen Computer, auf dem SPS-Programme ausgeführt werden. Auf einem Laufzeitcomputer muss Folgendes installiert sein:

- TwinCAT 3.1.4024.0 XAR (Runtime) oder höher
- TwinCAT 3 Function TF6510
- Lizenzen für TC1200 PLC und TF6510
- Für Testzwecke kann eine 7-Tage Testlizenz verwendet werden (siehe [Lizenzierung \[► 20\]](#))

Entwicklungs- und Laufzeitumgebung auf einem Computer

Wenn ein Computer als Laufzeit- und Entwicklungsumgebung verwendet werden soll (z. B. um ein SPS-Programm zu testen, bevor es in die Laufzeitumgebung geladen wird), müssen folgende Anforderungen erfüllt sein:

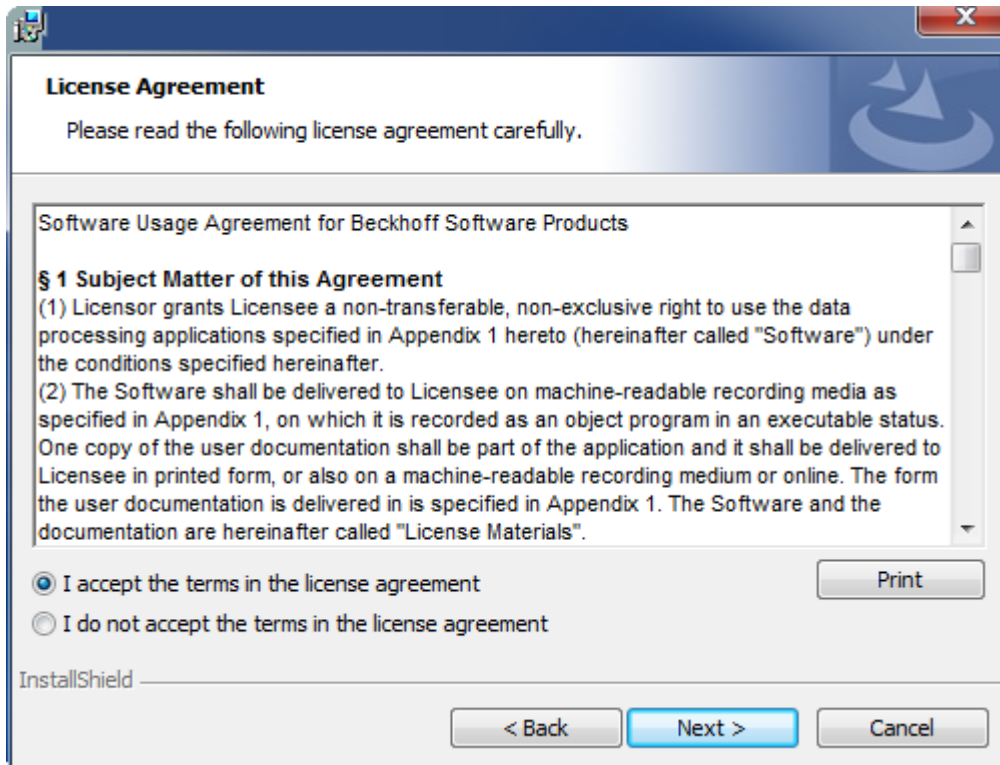
- TwinCAT 3.1.4024.0 XAE (Engineering) oder höher
- TwinCAT 3 Function TF6510
- Lizenzen für TC1200 PLC und TF6510
- Für Testzwecke kann eine 7-Tage Testlizenz verwendet werden (siehe [Lizenzierung \[► 20\]](#))

3.2 Installation

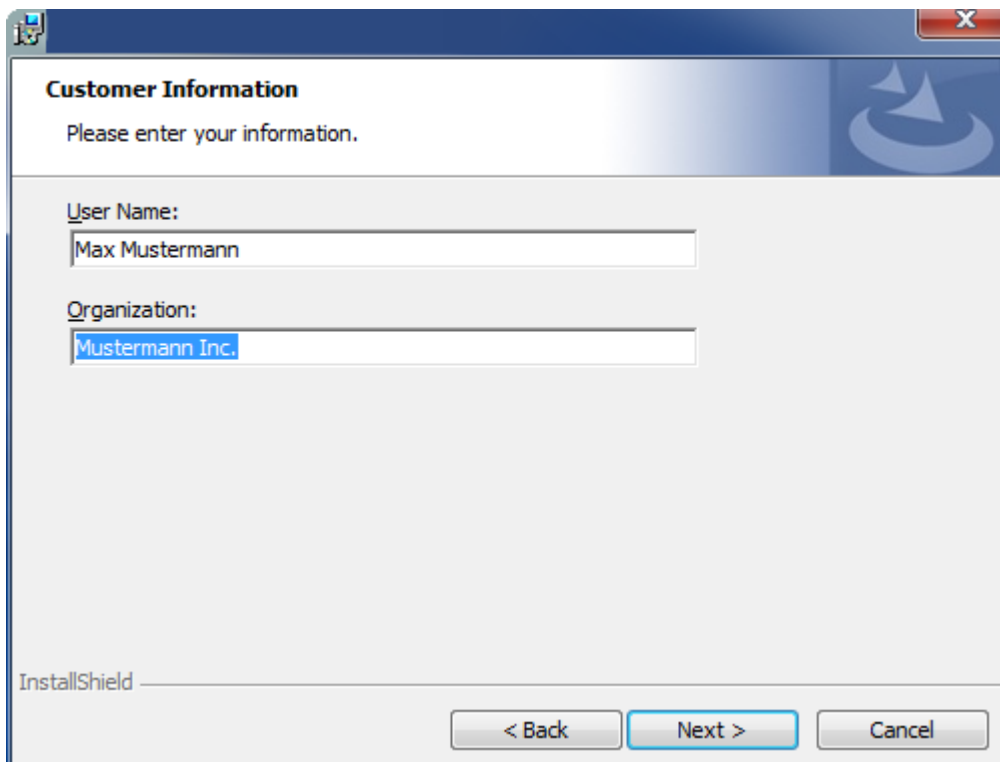
Nachfolgend wird beschrieben, wie die TwinCAT 3 Function für Windows-basierte Betriebssysteme installiert wird.

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

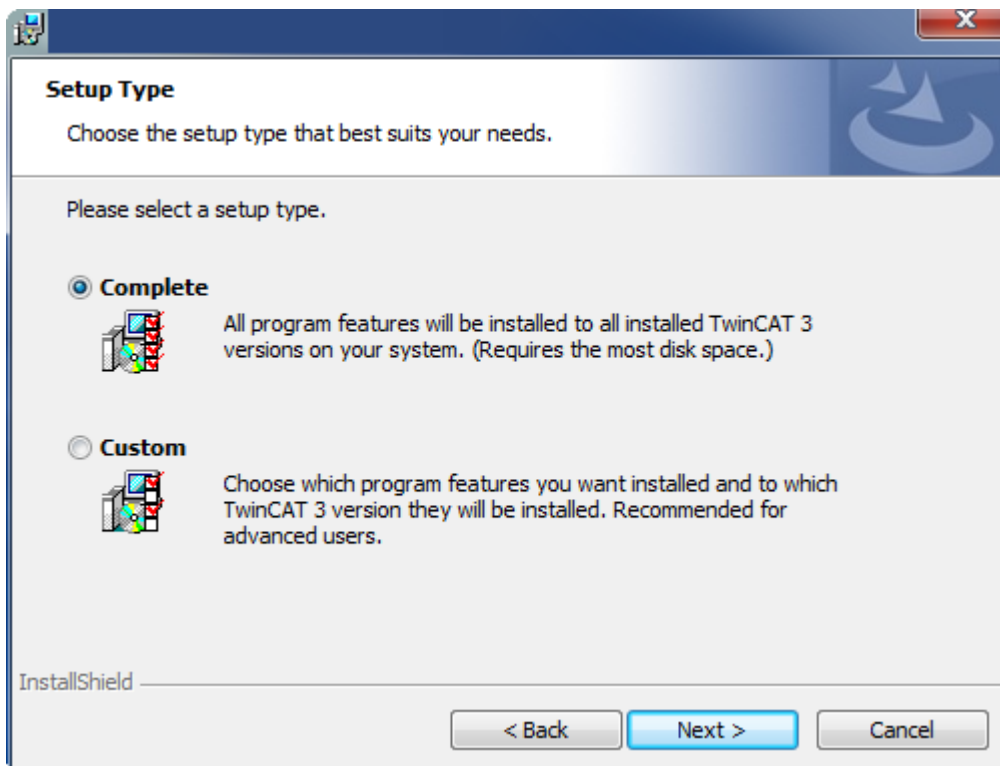
2. Akzeptieren Sie die Endbenutzerbedingungen und klicken Sie auf **Next**.



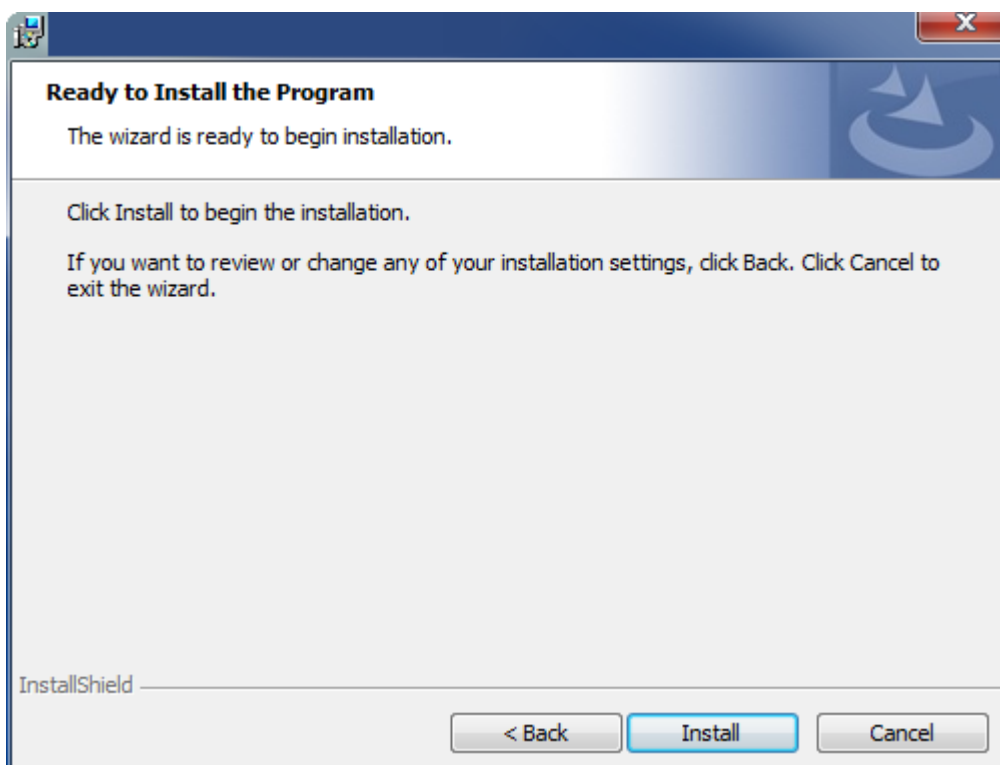
3. Geben Sie Ihre Benutzerdaten ein.



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

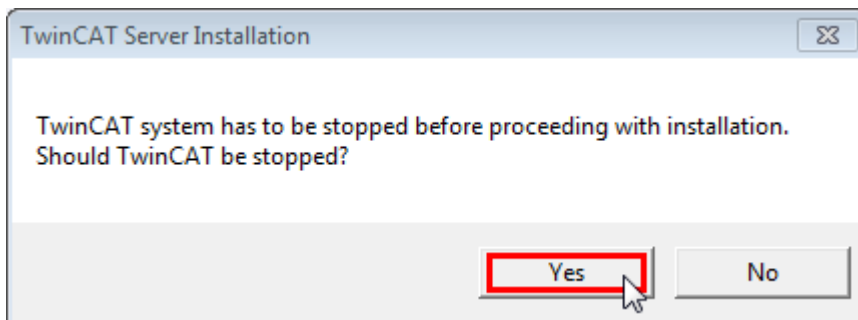


5. Wählen Sie **Next** und anschließend **Install**, um die Installation zu beginnen.

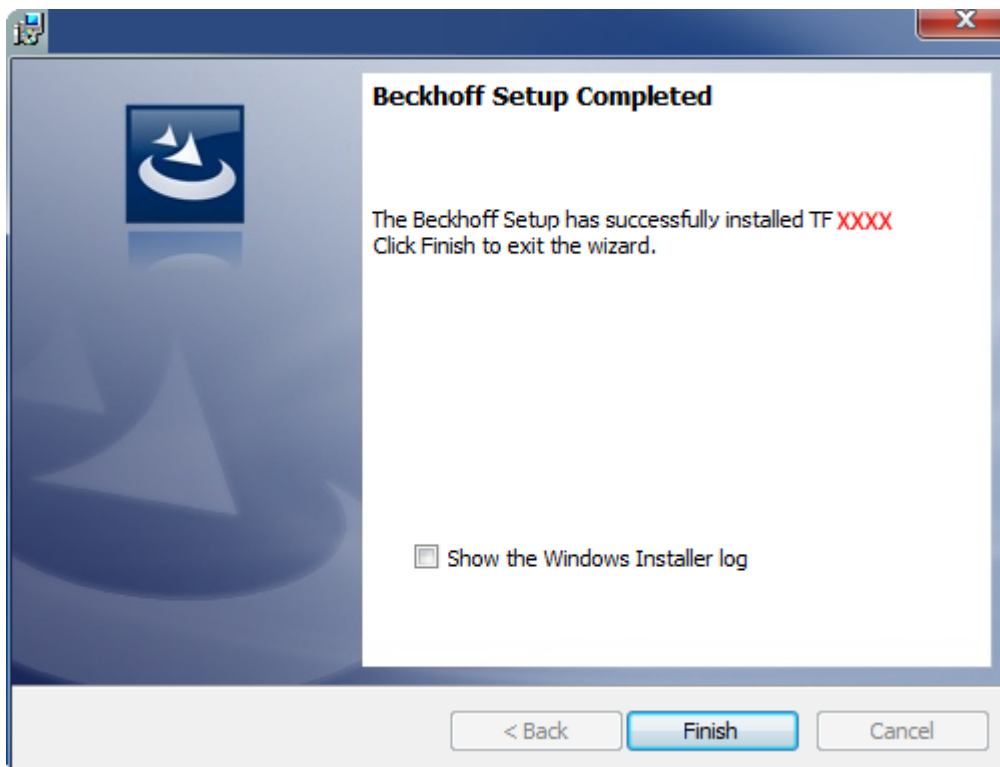


- ⇒ Ein Dialog weist Sie darauf hin, dass das TwinCAT-System für die weitere Installation gestoppt werden muss.

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



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



⇒ Die TwinCAT 3 Function wurde erfolgreich installiert und kann lizenziert werden (siehe [Lizenzierung](#) [► 20]).

3.3 Installation Windows CE

Nachfolgend wird beschrieben, wie eine TwinCAT 3 Function (TFxxx) auf einem Beckhoff Embedded-PC mit Windows CE installiert wird.

1. [Download der Setup-Datei und Installation](#) [► 18]
2. [CAB-Datei auf das Windows-CE-Gerät übertragen](#) [► 19]
3. [CAB-Datei auf dem Windows-CE-Gerät ausführen](#) [► 19]

Wenn bereits eine ältere TFxxx-Version auf dem Windows-CE-Gerät installiert ist, kann diese aktualisiert werden:

- [Upgrade der Software](#) [► 19]

Download der Setup-Datei und Installation

Die CAB-Installationsdatei für Windows CE ist Teil des TFxxx-Setups. Dieses wird Ihnen auf der Beckhoff-Homepage www.beckhoff.com zur Verfügung gestellt und enthält automatisch alle Versionen für Windows XP, Windows 7 und Windows CE (x86 und ARM).

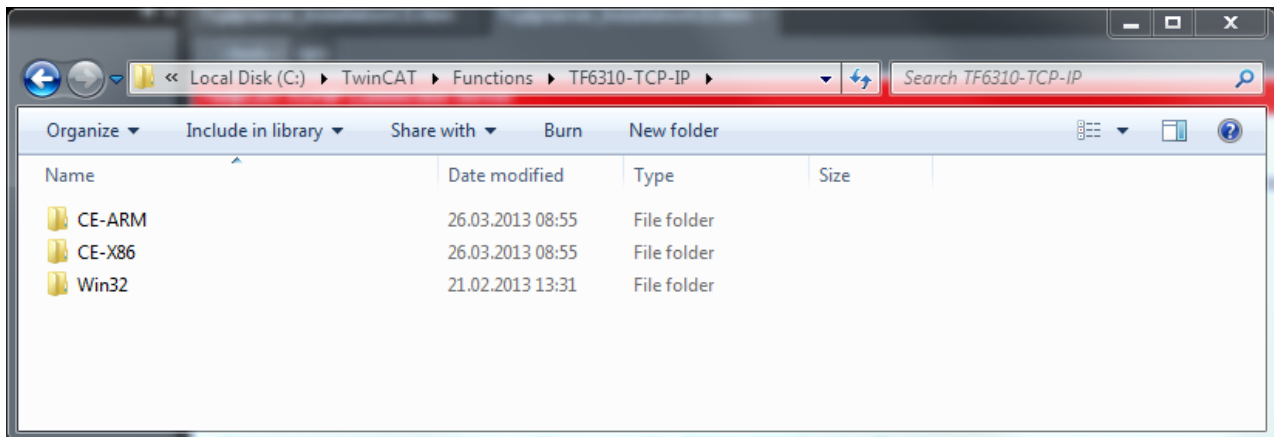
Laden Sie das TFxxx-Setup herunter und installieren Sie die TwinCAT 3 Function wie es im Abschnitt [Installation \[► 15\]](#) beschrieben wird.

Nach der Installation enthält der Installationsordner drei Verzeichnisse (pro Hardware-Plattform ein Verzeichnis):

- **CE-ARM:** ARM-basierte Embedded-PC, die unter Windows CE laufen, z. B. CX8090, CX9020
- **CE-X86:** X86-basierte Embedded-PC, die unter Windows CE laufen, z. B. CX50xx, CX20x0
- **Win32:** Embedded-PC, die unter Windows XP, Windows 7 oder Windows Embedded Standard laufen

Die Verzeichnisse CE-ARM und CE-X86 enthalten die CAB-Dateien der TwinCAT 3 Function für Windows CE in Bezug auf die jeweilige Hardware-Plattform des Windows-CE-Gerätes.

Beispiel: Installationsordner „TF6310“



CAB-Datei auf das Windows-CE-Gerät übertragen

Übertragen Sie die entsprechende CAB-Datei auf das Windows-CE-Gerät.

Für die Übertragung der ausführbaren Datei stehen Ihnen verschiedene Möglichkeiten zur Verfügung:

- über Netzwerkfreigaben
- über den integrierten FTP-Server
- über ActiveSync
- über CF/SD-Karten

Weitere Informationen finden Sie im Beckhoff Information System in der Dokumentation „Betriebssysteme“ (Embedded-PC > Betriebssysteme > [CE](#)).

CAB-Datei auf dem Windows-CE-Gerät ausführen

Nachdem Sie die CAB-Datei auf das Windows-CE-Gerät übertragen haben, führen Sie die Datei dort mit einem Doppelklick aus. Bestätigen Sie den Installationsdialog mit **OK**. Starten Sie das Windows-CE-Gerät anschließend neu.

Nach dem Neustart des Gerätes werden die Dateien der TwinCAT 3 Function (TFxxxx) automatisch im Hintergrund geladen und sind verfügbar.

Die Software wird in dem folgenden Verzeichnis auf dem Windows-CE-Gerät installiert:
`Hard Disk\TwinCAT\Functions\TFxxxx`

Upgrade der Software

Wenn auf dem Windows-CE-Gerät bereits eine ältere Version der TwinCAT 3 Function installiert ist, führen Sie die folgenden Schritte auf dem Windows-CE-Gerät durch, um ein Upgrade auf eine neue Version durchzuführen:

1. Öffnen Sie den CE Explorer, indem Sie auf **Start > Run** klicken und „Explorer“ eingeben.

2. Navigieren Sie nach *Hard Disk\TwinCAT\Functions\TFxxx\xxxx*.
 3. Benennen Sie die Datei *Tc*.exe* in *Tc*.old* um.
 4. Starten Sie das Windows-CE-Gerät neu.
 5. Übertragen Sie die neue CAB-Datei auf das Windows-CE-Gerät.
 6. Führen Sie die CAB-Datei auf dem Windows-CE-Gerät aus und installieren Sie die neue Version.
 7. Löschen Sie die Datei *Tc*.old*.
 8. Starten Sie das Windows-CE-Gerät neu.
- ⇒ Nach dem Neustart ist die neue Version aktiv.

3.4 Lizenzierung

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

Lizenzierung der Vollversion einer TwinCAT 3 Function

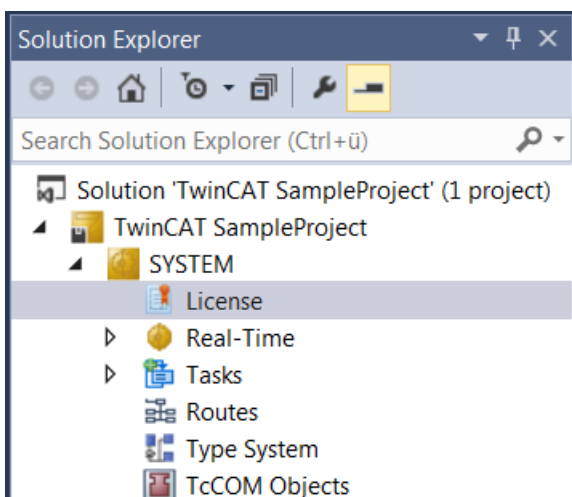
Die Beschreibung der Lizenzierung einer Vollversion finden Sie im Beckhoff Information System in der Dokumentation „[TwinCAT 3 Lizenzierung](#)“.

Lizenzierung der 7-Tage-Testversion einer TwinCAT 3 Function



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

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



⇒ Der TwinCAT-3-Lizenzmanager öffnet sich.

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

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

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

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

- Geben Sie den Code genauso ein, wie er angezeigt wird, und bestätigen Sie ihn.
- Bestätigen Sie den nachfolgenden Dialog, der Sie auf die erfolgreiche Aktivierung hinweist.

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

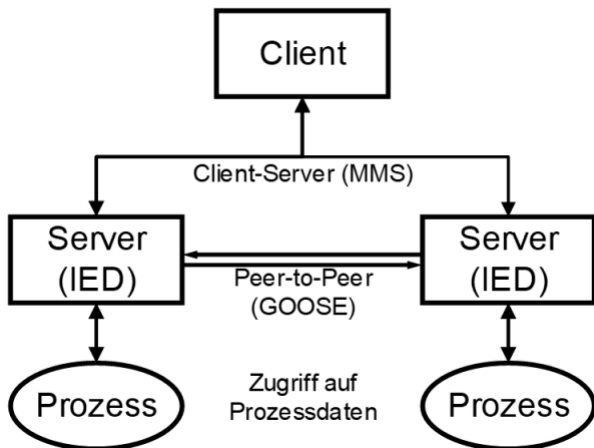
10. Starten Sie das TwinCAT-System neu.

⇒ Die 7-Tage-Testversion ist freigeschaltet.

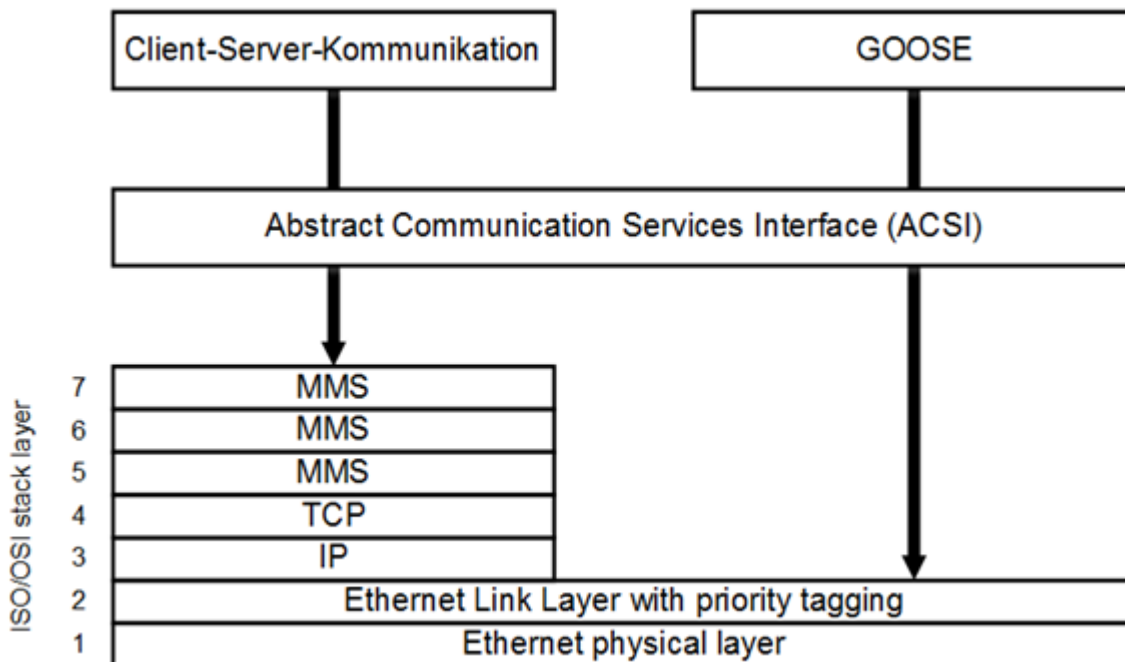
4 Technische Einführung

4.1 Normübersicht

Die IEC 61850 ist ein allgemeines Übertragungsprotokoll für die Schutz- und Leittechnik in elektrischen Schaltanlagen der Mittel- und Hochspannungstechnik (Stationsautomatisierung). Physikalisch basiert diese Kommunikation auf der Ethernet Technologie. Derzeit wird in TwinCAT die Server-Client-Kommunikation über MMS (Manufacturing Message Specification) und GOOSE unterstützt.



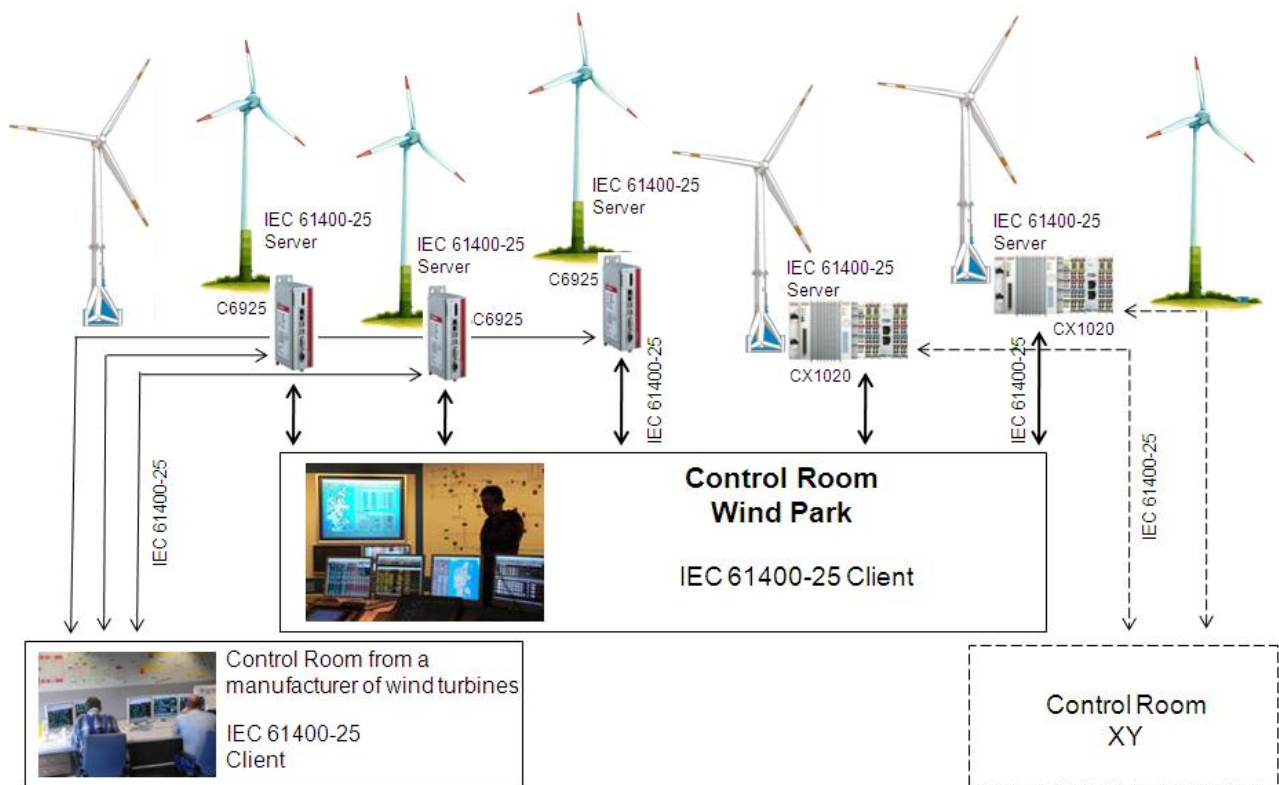
Der Kommunikationsstack, wie er in der unteren Abbildung gezeigt wird, ist Schicht für Schicht in die TwinCAT SPS in Form von Bibliotheken implementiert worden.



Die im Jahr 2004 veröffentlichte IEC 61850 ist strikt objektorientiert und ihr Datenmodell ist hierarchisch aufgebaut. Die verschiedenen Datenmodelle werden im folgenden Kapitel erläutert.

Anwendungsbeispiel

In diesem Fall ermöglicht die Verwendung der IEC 61400-25 die Kommunikation aller in einem Windpark befindlichen Windkraftanlagen verschiedener Hersteller mit einer Zentralstation. Die Standardisierung vermeidet die Verwendung herstellerspezifischer Protokolle, welche zu einem erhöhten Applikationsaufwand führen. Die Firma BECKHOFF unterstützt mit der Implementierung der IEC 61850 bzw. IEC 61400-25 in der TwinCAT Automationsuite diesen Weg.

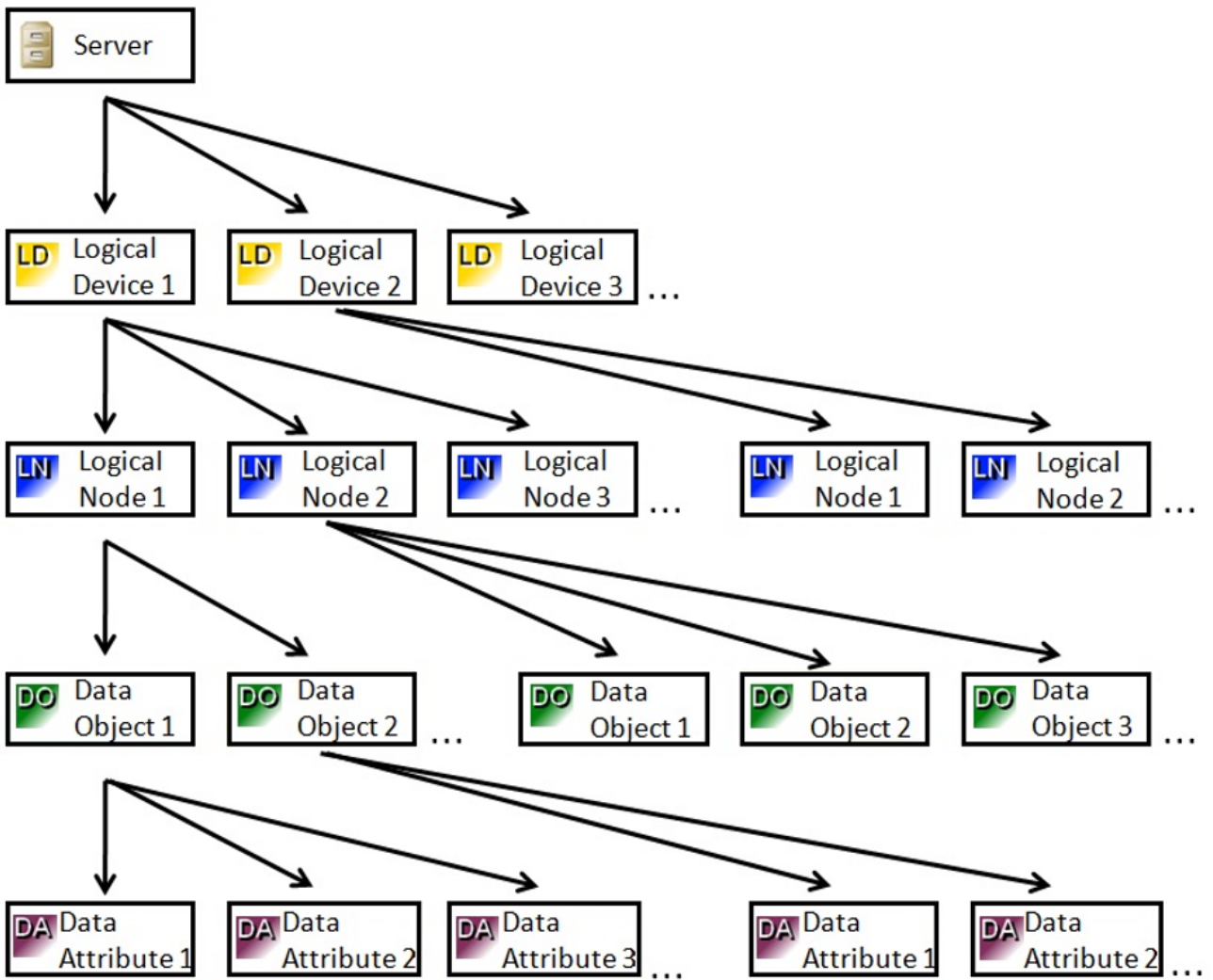


4.2 Datenmodell






Die IEC 61850 ist die Basisnorm für verschiedene Anwendungs-Spezialisierungen. Jede Spezialisierung definiert ihr eigenes Datenmodell bzw. erweitert das Basis-Datenmodell der IEC 61850.

Standard	Datenmodelle
IEC 61850	Basis Datenmodell für die Schutz- und Leittechnik in elektrischen Schaltanlagen
IEC 61850-7-410 (alt: IEC 62344)	Datenmodelle für die Kommunikation in Wasserkraftwerken
IEC 61850-7-420 (alt: IEC 62350)	Datenmodelle für verteilte/dezentrale Energieerzeugung und Speicherung
IEC 61400-25	Datenmodelle für die Kommunikation und Überwachung von Windkraftanlagen

Grundsätzlich gliedern sich die Datenmodelle in fünf Hierarchiestufen. Dabei geben die entsprechenden Standards alle nötigen Logical Nodes, Data Objects und Data Attributes vor. Je nach Anwendung müssen Logical Devices bzw. Server/Client an Hand der definierten Modelle konfiguriert werden. Im zugehörigen Beckhoff Produkt kann die Konfiguration im Tool TwinCAT Telecontrol Configurator [► 43] vorgenommen werden.



Hierarchiestufen

 Server	<p>Die erste Hierarchieebene im Datenmodell der IEC 61850 bzw. IEC 61400-25 bildet der Server. Der Server bietet für ein Gerät einen Anschlusspunkt an, wodurch entsprechende auf Ethernet basierende Kommunikationssysteme angebunden werden können. In diesem Fall würde TwinCAT, mit den zum Protokoll zugehörigen Bibliotheken, auf einem Industrie PC oder Embedded Gerät einen solchen Server darstellen.</p>
 Logical Device	<p>Die Logical Devices bilden die zweite Hierarchieebene im Datenmodell der IEC 61850 und IEC 61400-25. Diese Ebene unterteilt ein einzelnes physikalisches Gerät in mehrere separate Teile, so genannte Logical Devices (Logisches Gerät). Der Vorteil dieser Unterteilung ist, dass zusammengehörige Funktionen, beziehungsweise Objekte, aufgrund ihrer Gemeinsamkeiten zusammenstehen und beispielsweise gemeinsam in einen anderen Betriebsmodus geschaltet werden können. Dabei können je nach Gerät verschiedene Logical Nodes aus der dritten Hierarchieebene implementiert werden.</p>
 Logical Node	<p>Die Logical Nodes (logische Knoten) bilden die dritte Hierarchiestufe im Datenmodell der IEC 61850 bzw. IEC 61400-25. Sie repräsentieren die Informationen aller erdenklichen Teilfunktionen, die in der digitalen Stationsleittechnik und unter anderem in Windkraftanlagen vorkommen können. Dies sind sämtliche Schutzfunktionen, aber auch Automatisierungsfunktionen und Funktionen zur Verarbeitung von Mess- und Zählwerten. Für alle genormten Logical Nodes der IEC 61850 und IEC 61400-25 gibt es definierte Bezeichner, die immer aus einer Abkürzung mit vier Buchstaben bestehen. Die Abkürzung <i>XSWI</i> steht zum Beispiel für <i>Circuit Switch</i>. Ein Logical Device kann aus mehreren solchen Knoten bestehen.</p>
 Data Object	<p>Ein Logical Node kann aus mehreren Data Objects (Datenobjekte) bestehen, diese bilden die vierte Hierarchieebene des IEC 61850 bzw. IEC 61400-25 Datenmodells. Eine Besonderheit dieser Hierarchiestufe ist, dass Datenobjekte ihrerseits hierarchisch verschachtelt sein können. So kann ein Datenobjekt aus mehreren Unterdatenobjekten bestehen, bevor letztendlich auf einen Basis- oder Komplexen-Datentyp in einer Common Data Class (CDC) zugegriffen wird.</p>
 Data Attribute	<p>Die unterste Hierarchieebene der IEC 61850 und IEC 61400-25 bilden die Datenattribute. Sie stellen die Detailinformationen oder Werte der Datenobjekte dar. Da für viele Datenobjekte immer derselbe Umfang an Datenattributen definiert ist, werden die Attribute eines Objektes in so genannten Common Data Classes (CDC) zusammengefasst. Dabei wird zu jedem Datenobjekt die zugehörige CDC angegeben, die Datenobjekte sind also Spezialisierungen der CDC.</p>

4.3 Unterstützte Services

Generell unterstützt TF6510 Server und Client, sowie GOOSE nach IEC 61850. Hier finden Sie eine Übersicht über die von TwinCAT unterstützten IEC 61850 Services.

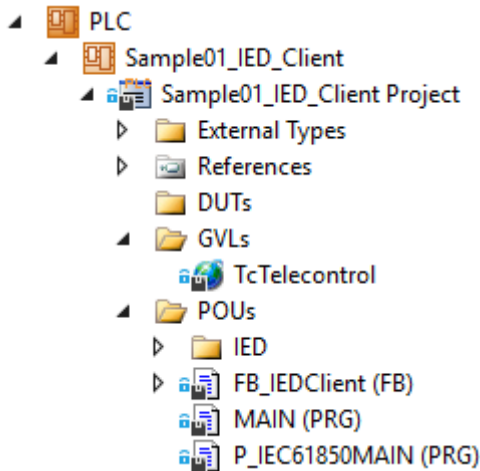
Service model	Supported services
Server (Device)	GetServerDirectory
Application association	Associate Abort Release
Logical device	GetLogicalDeviceDirectory
Logical node	GetLogicalNodeDirectory GetAllDataValues
Data	GetDataValues SetDataValues GetDataDefinition GetDataDirectory
Data set	GetDataSetValues DataSetValues GetDataSetDirectory CreateDataSet DeleteDataSet
Reporting	Buffered RCB: Report GetBRCBValues SetBRCBValues Unbuffered RCB: Report GetURCBValues SetURCBValues
Generic substation events (GSE)	GOOSE support for publisher and subscriber GOOSE CB: SendGOOSEMessage GetGoCBValues SetGOCBValues
Control	Select SelectWithValue Cancel Operate CommandTermination TimeActivatedOperate

4.4 Allgemeine Client - Projektstruktur

Jedes der Client-Beispielprojekte folgt einer allgemeinen Struktur, welche in diesem Dokument erläutert werden soll. Zu dieser Struktur gehört die Verschachtelung der Funktionsbausteine im Solution-Explorer und der Aufbau der StateMachine im Client-Baustein, der die Verbindung zum Server herstellt und den Datenaustausch realisiert. Die Grundstruktur der TwinCAT IEC 61850 Beispielprojekte basiert auf der automatisch vom TwinCAT Telecontrol Configurator generierten SPS-Projektstruktur. Der TwinCAT Telecontrol Configurator verwendet bei der Generierung der Solution wiederum das „TwinCAT XAE Project (XML format)“ als Vorlage. Die in den Beispielen kommunizierten Datenmodelle spiegeln unterschiedliche

IEC 61850 Server wider. Sie unterscheiden sich von Beispiel zu Beispiel und liegen dem Quellcode als icd-Dateien bei. Die ICD-Dateien können auch von einer Third-Party-Software zur Simulation eines Servers verwendet werden.

Zuerst eine kurze Betrachtung des Aufbaus eines TwinCAT IEC 61850 SPS-Client-Projekts:



Der generierte Solution-Name (wenn nicht anders angegeben) entspricht dem TwinCAT Telecontrol Configurator Projektnamen. Der automatisch generierte TwinCAT SPS-Projektname (wenn nicht anders angegeben) hat wiederum folgenden Aufbau: „[Projektname]_[IEDName]_Client“.

Standardmäßig besitzt jedes Beispielprojekt einen „DUTs“, „GVLs“ und „POUs“ Ordner. In dem „GVLs“-Ordner ist eine globale Variablenliste (GVL) mit dem Namen: „TcTelecontrol“ hinterlegt. In dieser globalen Variablenliste (siehe Quellcode unten) werden folgende Funktionsbausteine instanziiert und initialisiert:

- Eine Client-Bausteininstanz des Typs: FB_[IEDName]Client (Verbindungsmanagement und Datenaustausch).
- Eine IED-Datenmodell-Bausteininstanz des Typs: FB_IED_[IEDName].
- Optional (abhängig von der GOOSE-Subscriber-Konfiguration): Eine oder mehrere Bausteininstanzen des Typs: „FB_[IEDName]Gse“ für GOOSE-Kommunikation und GSE-Management.

Außerdem wird dort auch die während der Codegenerierung verwendete Codeversion und die verwendete Version des TwinCAT Telecontrol Configurator abgelegt.

Namensraum: TcTelecontrol

Typ: Globale Variablenliste (GVL)

```
VAR_GLOBAL
  ipCreator      : I_AcsiCodeCreatorClass := GVL_AcsiVars.Creator.SetCodeRev(codeRev:=2).SetGui
  Ver(major:=1, minor:=0, build:=93, revision:=10);
  fb[IEDName]    : FB_IED_[IEDName];
  fb[IEDName]Client : FB_[IEDName]Client := (fbConnection:=(ipIED:=fb[IEDName], settings:=(sRemoteHost:='127.0.0.1')));
  fb[IEDName]Gse : FB_[IEDName]Gse := (fbAdapter:=(ipIED:=fb[IEDName], settings:=(sMulticastAddr:='01-0C-CD-01-00-00', eDispatchMode:=E_GseDispatchMode.NonPromiscuous)));
END_VAR
```

Im „POUs“-Ordner befindet sich ein weiterer Ordner: „[IEDName]“, der die gesamte hierarchische Struktur des IED-Datenmodells als Funktionsbausteine beinhaltet. In diesem Ordner befindet sich auch der IED-Baustein, der in der globalen Variablenliste instanziiert wird und bereits erwähnt wurde. Auf der gleichen Ebene liegt der Client-Baustein: „FB_[IEDName]Client“, welcher den Verbindungsaufbau und den Datenaustausch mit einem IEC 61850 Server realisiert. Das TwinCAT SPS-Projekt enthält standardmäßig ein „MAIN“-Programm. Dieses wird zyklisch von einer TwinCAT Task aufgerufen und ruft wiederum das Programm „P_IEC61850MAIN“ auf. Das Programm „P_IEC61850MAIN“ kapselt den Aufruf des Client-Bausteins und des optionalen GSE-Bausteins, trennt die IEC 61850-Kommunikation von dem restlichen SPS-Maschinenprogramm und hilft z. B. bei der Implementierung weiterer Clients.

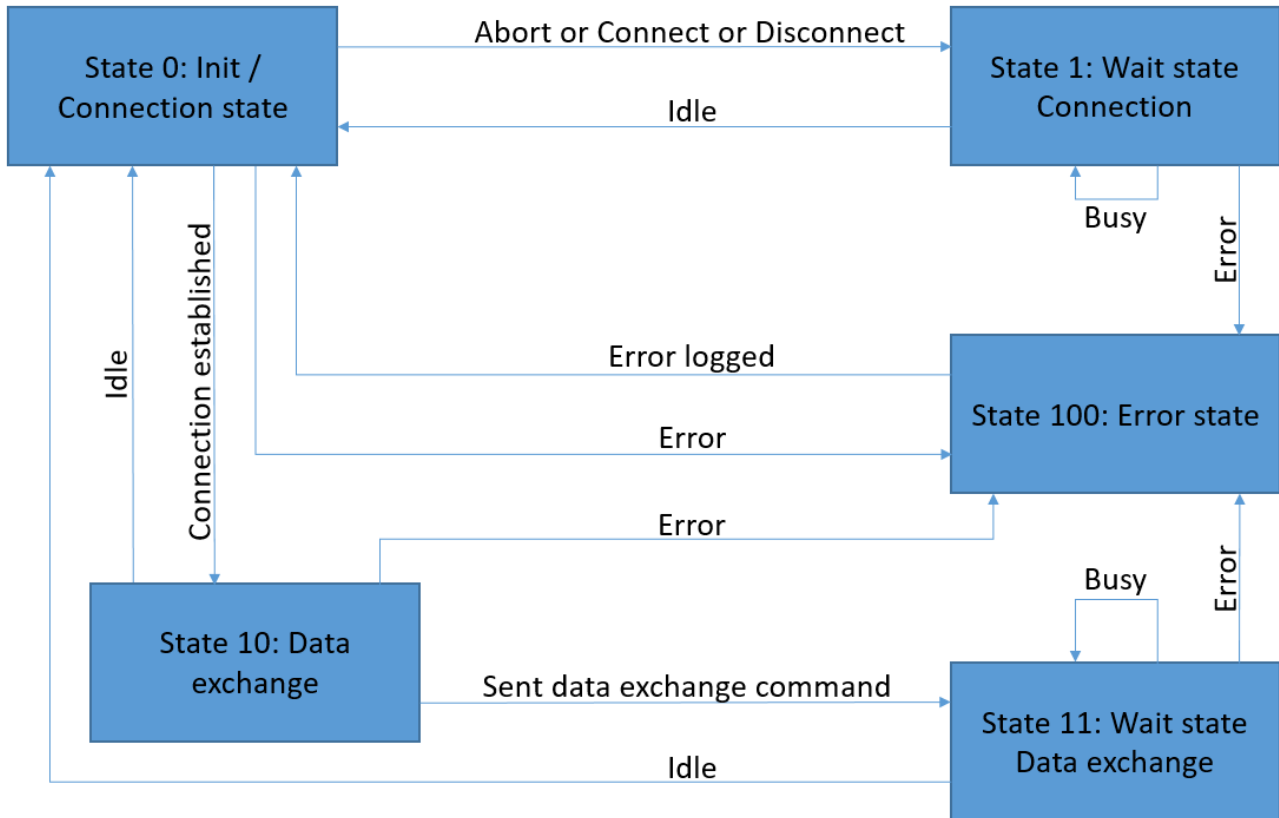
```
PROGRAM MAIN
VAR
END_VAR
P_IEC61850MAIN();
```



```
PROGRAM P_IEC61850MAIN
VAR
END_VAR
```

```
fb[IEDName]Client();
fb[IEDName]Gse();
```

In dem FB_[IEDName]Client-Baustein befindet sich eine State machine, von der die grundlegenden Zustände in jedem Client-Beispiel genutzt werden. Diese Zustände sind im folgenden Diagramm grafisch dargestellt:



State 0 (Init state): Die State machine befindet sich in diesem Zustand, sobald das SPS-Programm gestartet wurde. Hier (und im State 1) werden Befehle zur Verwaltung der Client-Server-Verbindung behandelt. Gesteuert wird dies hauptsächlich über vier boolesche Variablen. Diese Variablen, wenn sie gesetzt sind, aktivieren die entsprechenden Befehle (in diesem Fall sind es einmalige Methodenaufrufe am Client-Baustein).

- **_bAbort:** Ruft die Methode „AbortReq“ auf, welche den Befehl zum Abbruch der Client-Verbindung zum Server aktiviert.
- **_bConnect:** Ruft die Methode „AssociateReq“ auf, welche den Befehl zum Aufbau einer neuen Client-Verbindung zum Server aktiviert.
- **_bDisconnect:** Ruft die Methode „ReleaseReq“ auf, welche den Befehl zum geregelten Abbau einer bestehenden Client-Verbindung zum Server aktiviert.
- **_bReconnect:** Ruft ebenfalls die Methode „AssociateReq“ auf, falls die Client-Verbindung zum Server abgebrochen/getrennt wurde, aber automatisch wiederhergestellt werden soll.

Die zuvor aufgezählten Methoden, welche in diesem State einmalig aufgerufen werden, benötigen für die Abarbeitung länger als einen SPS-Zyklus. Aus diesem Grund wechselt die State machine in einen Warte-Zustand (State 1), in dem auf die Beendigung des aktivierten Befehls gewartet wird.

Falls die Client-Verbindung zum Server zuvor bereits hergestellt wurde, wird die State machine in den Zustand zur Datenübertragung wechseln (State 10).

State 1 (Wait State): In diesem Zustand verweilt der Client bis die Befehlsabarbeitung zur Verwaltung der Client-Server-Verbindung nicht mehr beschäftigt (busy) ist. Solange die Verbindung hergestellt, abgebaut oder abgebrochen wird, befindet sich die State machine im State 1. Sobald der Befehl erfolgreich abgearbeitet wurde, wird die State machine zurück in den State 0 (Init State) versetzt.

State 10 (Data exchange): Wenn die Statemachine in diesem Zustand ist, dann ist die Client-Verbindung zum Server bereits erfolgreich hergestellt worden. Der Client ist bereit für die Datenübertragung zum Server. Während der Datenübertragung werden Befehle zum Versenden oder Empfangen der Daten aktiviert.

In diesem Zustand unterscheiden sich die Beispielprojekte untereinander. Abhängig von der gewünschten Funktionalität oder Logik in der Applikation werden hier unterschiedliche Methoden oder Hilfsbausteine aufgerufen. Außerdem kann der Client-Baustein um weitere States erweitert werden.

Sämtliche Methodenaufrufe, die Befehle zur Datenübertragung aktivieren, benötigen für eine erfolgreiche Abarbeitung mehrere SPS-Zyklen, daher muss die Statemachine, nach dem Aufruf einer solchen Methode, in den State 11 (Wait state) versetzt werden.

Bei einer aktiven Verbindung und keiner Datenübertragung wechselt die Statemachine zwischen dem State 0 und 10. Die Statemachine wird wieder in den State 0 versetzt, um auf Änderungen im Client-Server-Verbindungsstatus zu reagieren und sie im State 0 zu behandeln.

State 11 (Wait state): Dieser Zustand ist ein weiterer Warte-Zustand. Sobald ein Befehl zur Datenübertragung (aktiviert im State 10) abgearbeitet wurde, wird die Statemachine in den State 11 versetzt und wartet bis die Befehlsabarbeitung nicht mehr beschäftigt (busy) ist. Danach wechselt die Statemachine in den State 0.

State 100 (Error state): Sobald ein Fehler bei der Aktivierung oder Abarbeitung eines Befehls auftritt, wird die Statemachine in den State 100 versetzt. Hier wird der Fehler geloggt und die Statemachine zurück in den State 0 versetzt.

```

FUNCTION_BLOCK FB_ [IEDName]Client
VAR_INPUT
    fbConnection          : FB_iec61850ClientClass;
END_VAR
VAR
    _bAbort              : BOOL;
    _bDisconnect         : BOOL;
    _bConnect            : BOOL;
    _bReconnect          : BOOL := TRUE;
    _bReadAllData        : BOOL := TRUE;
    state                : BYTE;
    eState               : E_AsyncEnvironmentState;
    bBusy                : BOOL;
    bSuccess             : BOOL;
    ipResult             : I_AsyncServiceResultClass;
    sLastErrorResult    : T_MaxString;
    fbAbortReason        : FB_ServiceErrorClass := (stError:=SUCCESS_EVENT);
    sLastAbortReason    : T_MaxString;
    nInvokeID           : UDINT;
    eServiceError        : E_AcsiServiceError;
    nServiceError        : UDINT;
    nCmdError            : UDINT;
    sObjReference        : T_AcsiObjectReference;
    sCtrlReference       : T_AcsiObjectReference;

    bGetServerDirectory : BOOL := TRUE;
    bGetLogicalDeviceDirectory: BOOL := TRUE;
    bGetLogicalNodeDirectory : BOOL := TRUE;
    bGetAllServerValues   : BOOL := TRUE;
    bGetAllServerValues   : BOOL := TRUE;
    bGetAllDataValues_LLNO_ST : BOOL := TRUE;
    bGetDataValues_LLNO_ST_Beh: BOOL := TRUE;
END_VAR

fbConnection.Execute();
eState:= fbConnection.eState;

CASE state OF
    0:
        IF _bAbort THEN
            _bAbort:= FALSE;
            bSuccess:= fbConnection.AbortReq(ipReason:=fbAbortReason, ipSink:=0, ipResult=>ipResult)
;
            state:= SEL(bSuccess, 100, 1);
        ELSIF eState = E_AsyncEnvironmentState.Idle AND (_bConnect OR _bReconnect) THEN
            _bConnect:= FALSE;
            bGetAllServerValues:= SEL(_bReadAllData, bGetAllServerValues, TRUE);
            bSuccess:= fbConnection.AssociateReq(ipSink:=0, ipResult=>ipResult);
            state:= SEL(bSuccess, 100, 1);
        ELSIF eState = E_AsyncEnvironmentState.Established AND _bDisconnect THEN
            _bDisconnect:= FALSE;

```

```

        bSuccess:= fbConnection.ReleaseReq(ipSink:=0, ipResult=>ipResult);
        state:= SEL(bSuccess, 100, 1);
    ELSIF eState = E_AsyncEnvironmentState.Established THEN
        state:= 10;
    END_IF
    _bConnect:= FALSE;
    _bDisconnect:= FALSE;
1:
    IF ipResult <> 0 THEN
        ipResult.Execute();
        IF NOT (bBusy:=ipResult.IsBusy()) THEN
            state:= SEL(ipResult.IsCompleted(), 100, 0);
        END_IF
    END_IF
10:
    IF bGetServerDirectory THEN
        bGetServerDirectory:= FALSE;
        bSuccess:= fbConnection.GetServerDirectoryReq(ipServer:=fb[IEDName], eClass:=E_AcsiServerDirectoryClass.LogicalDevice, hUser:=0, ipSink:=0, nInvokeID=>nInvokeID, ipResult=>ipResult);
        state:= SEL(bSuccess, 100, 11);
    ELSIF bGetLogicalDeviceDirectory THEN
        bGetLogicalDeviceDirectory:= FALSE;
        bSuccess:= fbConnection.GetLogicalDeviceDirectoryReq(ipLogicalDevice:=fb[IEDName].IEDLD1, hUser:=0, ipSink:=0, nInvokeID=>nInvokeID, ipResult=>ipResult);
        state:= SEL(bSuccess, 100, 11);
    ELSIF bGetLogicalNodeDirectory THEN
        bGetLogicalNodeDirectory:= FALSE;
        bSuccess:= fbConnection.GetLogicalNodeDirectoryReq(ipLogicalNode:=fb[IEDName].IEDLD1.LLN0, eClass:=E_AcsiLogicalNodeClass.DataSet, hUser:=0, ipSink:=0, nInvokeID=>nInvokeID, ipResult=>ipResult);
        state:= SEL(bSuccess, 100, 11);
    ELSIF bGetAllServerValues THEN
        bGetAllServerValues:= FALSE;
        bSuccess:= fbConnection.GetAllServerValuesReq(ipServer:=fb[IEDName], hUser:=0, ipSink:=0, nInvokeID=>nInvokeID, ipResult=>ipResult);
        state:= SEL(bSuccess, 100, 11);
    ELSIF bGetAllDataValues_LLNO_ST THEN
        bGetAllDataValues_LLNO_ST:= FALSE;
        bSuccess:= fbConnection.GetAllDataValuesReq(ipLogicalNode:=fb[IEDName].IEDLD1.LLN0, eFc:=E_AcsiFc.ST_, hUser:=0, ipSink:=0, nInvokeID=>nInvokeID, ipResult=>ipResult);
        state:= SEL(bSuccess, 100, 11);
    ELSIF bGetDataValues_LLNO_ST_Beh THEN
        bGetDataValues_LLNO_ST_Beh:= FALSE;
        bSuccess:= fbConnection.GetDataValuesReq(ipData:=fb[IEDName].IEDLD1.LLN0.Beh, eFc:=E_AcsiFc.ST_, hUser:=0, ipSink:=0, nInvokeID=>nInvokeID, ipResult=>ipResult);
        state:= SEL(bSuccess, 100, 11);
    ELSE
        state:= 0;
    END_IF
11:
    IF ipResult <> 0 THEN
        ipResult.Execute();
        IF NOT (bBusy:=ipResult.IsBusy()) THEN
            state:= SEL(ipResult.IsCompleted(), 100, 0);
        END_IF
    END_IF
100:
    state:= 0;
    IF ipResult <> 0 THEN
        nCmdError:= nCmdError + 1;
        sLastErrorResult:= ipResult.Dump();
    END_IF
END_CASE

```

GOOSE-Subscriber (optional)

TwinCAT Telecontrol Configurator kann in einem Client-Projekt, während der SPS-Code-Generierung den SPS-Code für einen GOOSE-Subscriber ebenfalls generieren (siehe Codebeispiel unten). Dies setzt voraus, dass der Anwender vorher die GOOSE-Komponenten wie z.B. GoCBs (goose control blocks) im TwinCAT Telecontrol Configurator erstellt oder aus einer SCL-Datei (z. B. ICD-Datei) importiert hat.

Standardmäßig wird bei der Codegenerierung ein Baustein mit dem Namen: „FB_[IEDName]Gse“ instanziiert und der globalen Variablenliste „TcTelecontrol“ hinzugefügt. Dieser Baustein stellt die Verbindung zwischen einem Netzwerkadapter des TwinCAT Steuerungsrechners, dem IED-Datenmodell und der GOOSE-Konfiguration in den GoCBs her. Die GoCBs werden in dem IED-Datenmodell (normalerweise im LLN0) instanziiert. Jeder GoCB besitzt ein Baustein-Unterelement mit dem Namen: „Subscriber“. Über die „Subscriber“-Methodenaufrufe kann die Subscription aus dem SPS-Code aktiviert („Enable“) bzw. deaktiviert werden („Disable“). Defaultmäßig wird die Subscription beim SPS-Programmstart

für alle GoCBs aktiviert. Dies wird durch die mit „TRUE“ initialisierte „bSubscriber“-Variable gesteuert. Über eine steigende Flanke an der „bUnsubscribe“-Variablen kann die Subscription für alle GoCBs deaktiviert werden. Die über diese Methoden abgesetzten Subscriber-Befehle werden sofort ausgeführt, ohne Wartezyklen oder weitere States die nötig sind, um die Befehlsabarbeitung abzuschließen. Der hier beschriebene „Subscriber“ liest die Konfiguration und aktualisiert zwar den Status des GoCBs (Attribute „GoEna“ wird z.B. auf „TRUE“ oder „FALSE“ gesetzt), er nutzt aber nicht die Client-Server-Dienste wie „SetGoCBValues“ oder „GetGoCBValues“, um den „Publisher“ auf der Serverseite zu aktivieren bzw. zu deaktivieren. D.h. der generierte Code implementiert bereits einen Subscriber der z. B. im ersten SPS-Zyklus oder aus dem SPS-Code zum beliebigen Zeitpunkt aktiviert bzw. deaktiviert werden kann. Die benötigten GoCB-Konfigurationseinstellungen (GoCB-Attributwerte) können über Initialisierungswerte vorgenommen werden. Die Konfiguration der GoCBs kann aber bereits in dem TwinCAT Telecontrol Configurator geschehen. Die Initialisierungswerte werden während der Codegenerierung automatisch generiert und zugewiesen. Wenn der Subscriber aktiviert wurde und die Konfiguration des GoCBs und des Netzwerkadapters zu dem empfangenden GOOSE-Frame eine Übereinstimmung ergibt, werden die GOOSE-Daten ins TwinCAT IED-Datenmodell umkopiert (gemappt). Die „Execute“-Methode muss in der übrigen Zeit zyklisch aufgerufen werden. Sie ist unter anderem für die Aktualisierung der Statusinformation im GoCB verantwortlich.

```

FUNCTION_BLOCK FB_[IEDName]Gse IMPLEMENTS I_GseLinkStatusEventSink
VAR_INPUT
    fbAdapter      : FB_GseAdapterClass := (ipLinkStatus:=THIS^);
END_VAR
VAR
    eLinkStatus   : E_GseLinkStatus;
    bSuccess       : BOOL;
    ipError        : I_ServiceErrorClass;
    bSubscribe     : BOOL := TRUE;
    bUnsubscribe   : BOOL;
END_VAR

bSuccess:= fbAdapter.Execute(ipError=>ipError);
IF bSubscribe THEN
    bSubscribe:= FALSE;
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb01.Subscriber.Enable(ipAdapter:=fbAdapter, ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb02.Subscriber.Enable(ipAdapter:=fbAdapter, ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb03.Subscriber.Enable(ipAdapter:=fbAdapter, ipError=>ipError);
ELSIF bUnsubscribe THEN
    bUnsubscribe:= FALSE;
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb01.Subscriber.Disable(ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb02.Subscriber.Disable(ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb03.Subscriber.Disable(ipError=>ipError);
ELSE
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb01.Subscriber.Execute(ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb02.Subscriber.Execute(ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb03.Subscriber.Execute(ipError=>ipError);
END_IF

```

Im Projektbaum unter dem Zweig I/O-Device finden Sie eine Netzwerkadapterinstanz mit dem Namen „GSE (RT-Ethernet Adapter)“. Diese Adapterinstanz muss entsprechend konfiguriert werden, d.h. die I/O-Konfiguration muss an die vorhandene Hardware und auf die Zielplattform, auf der das Projekt laufen soll, angepasst werden.

Eine erneute I/O-Konfiguration ist auch dann nötig, wenn Sie die Zielplattform wechseln. Diese Konfiguration muss manuell in TwinCAT XAE vorgenommen werden. Zusätzlich zu der I/O-Konfiguration des Netzwerkadapters muss ein Link zwischen dem Netzwerkadapter und den SPS-Bausteinen für die Goose-Kommunikation hergestellt werden. Mit dem Link können die vom Netzwerkadapter empfangenen Daten an die Instanz des Funktionsbausteins: „FB_[IEDName]Gse“ weitergeleitet werden. In umgekehrter Richtung kann die Instanz des Funktionsbausteins „FB_[IEDName]Gse“ die zu sendenden Daten an den Netzwerkadapter weiterleiten.

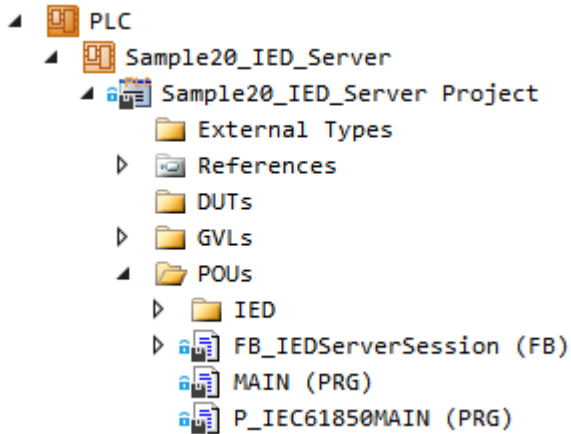
Hier finden Sie weitere Informationen: [RT – Ethernet Adapter Konfiguration \[► 37\]](#).

4.5 Allgemeine Server - Projektstruktur

Alle Server-Beispielprojekte folgen, ähnlich wie die Client-Beispielprojekte, ebenfalls einer allgemeinen Struktur. Im Gegensatz zum Client wird beim Server die Verbindung nicht aktiv Serverseits aufgebaut. Nach dem Programmstart wartet der Server auf einen Verbindungswunsch eines Clients und akzeptiert oder lehnt diesen ab. Außerdem muss es für den Server möglich sein, Verbindungen zu mehreren Clients aufzubauen.

Daraus ergibt sich eine andere Server-Projektstruktur als beim Client. Die Grundstruktur der TwinCAT IEC 61850 Server-Beispielprojekte basiert auf der automatisch vom TwinCAT Telecontrol Configurator generierten SPS-Projektstruktur. Der TwinCAT Telecontrol Configurator verwendet bei der Generierung der Solution wiederum das „TwinCAT XAE Project (XML format)“ als Vorlage. Die in den Beispielen kommunizierten Datenmodelle spiegeln unterschiedliche IEC 61850 Server wider. Sie unterscheiden sich von Beispiel zu Beispiel und liegen dem Quellcode als ICD-Dateien bei. Die ICD-Dateien können auch von einer Third-Party-Software zur Simulation verwendet werden.

Aufbau eines TwinCAT IEC 61850-SPS-Server-Projekts:



Der generierte Solution-Name (wenn nicht anders angegeben) entspricht dem TwinCAT Telecontrol Configurator Projektnamen. Der automatisch generierte TwinCAT SPS-Projektname (wenn nicht anders angegeben) hat wiederum folgenden Aufbau: „[Projektname]_[IEDName]_Server“.

Standardmäßig besitzt jedes Beispielprojekt einen „DUTs“, „GVLs“ und „POUs“ Ordner. In dem „GVLs“-Ordner ist eine globale Variablenliste (GVL) mit dem Namen: „TcTelecontrol“ hinterlegt. In dieser globalen Variablenliste (siehe Quellcode unten) werden folgende Funktionsbausteine instanziiert und initialisiert:

- Eine Server-Bausteininstanz des Typs: „FB_iec61850ServerClass“ (Server Konfigurationseinstellungen und Verbindungsmanagement).
- Mindestens eine Instanz des Server-Session-Bausteins des Typs: „FB_[IEDName]ServerSession[1..n]“ (Verbindungsmanagement einer einzelnen Verbindung und Datenaustausch mit einem Client). Fügen Sie an dieser Stelle manuell weitere Instanzen des Server-Session-Bausteins hinzu, falls der Server gleichzeitig mit mehr als nur einem Client kommunizieren soll.
- Eine IED-Datenmodell-Bausteininstanz des Typs: „FB_IED_[IEDName]“.
- Optional (abhängig von der GOOSE-Publisher-Konfiguration): Eine oder mehrere Bausteininstanzen des Typs: „FB_[IEDName]Gse“ für GOOSE-Kommunikation und GSE-Management.

Außerdem wird dort auch die, während der Codegenerierung verwendete Codeversion und die verwendete Version des TwinCAT Telecontrol Configurator abgelegt.

Namensraum: TcTelecontrol

Typ: Globale Variablenliste (GVL)

```

VAR_GLOBAL
    ipCreator          : I_AcsiCodeCreatorClass := GVL_AcsiVars.Creator.SetCodeRev(codeRev:=2)
).SetGuiVer(major:=1, minor:=1, build:=94, revision:=1);
    fb[IEDName]       : FB_IED_[IEDName];
    fb[IEDName]Server : FB_iec61850ServerClass := (ipIED:=fb[IEDName], settings:=(bEnable:=TRUE, sLocalHost:='127.0.0.1'));
    fb[IEDName]ServerSession1 : FB_[IEDName]ServerSession := (fbConnection:=(ipServer:=fb[IEDName]Server, settings:=(bEnable:=TRUE)));
    fb[IEDName]Gse    : FB_[IEDName]Gse := (fbAdapter:=(ipIED:=fb[IEDName], settings:=(sMulticastAddr:='01-0C-CD-01-00-00', eDispatchMode:=E_GseDispatchMode.NonPromiscuous)));
END_VAR
    
```

Im „POUs“-Ordner befindet sich ein weiterer Ordner: „[IEDName]“, der die gesamte hierarchische Struktur des IED-Datenmodells als Funktionsbausteine beinhaltet. In diesem Ordner befindet sich auch der IED-Datenmodell-Baustein, der in der globalen Variablenliste instanziiert wird und bereits erwähnt wurde. Auf der gleichen Ebene liegt der Server-Session-Baustein: „FB_[IEDName]ServerSession“, welcher den Verbindungsaufbau und den Datenaustausch mit einem IEC 61850 Client realisiert.

Das TwinCAT SPS-Projekt enthält standardmäßig ein „MAIN“-Programm. Dieses wird zyklisch von einer TwinCAT Task aufgerufen und ruft wiederum das Programm „P_IEC61850MAIN“ auf. Das Programm „P_IEC61850MAIN“ kapselt den Aufruf des Server-, Server-Session- und des optionalen Gse-Bausteins, trennt die IEC 61850-Kommunikation von dem restlichen SPS-Maschinenprogramm.

```
PROGRAM MAIN
VAR
END_VAR

P_IEC61850MAIN();

PROGRAM P_IEC61850MAIN
VAR
END_VAR

fb[IEDName]Server.Execute();
fb[IEDName]ServerSession1();
fb[IEDName]Gse();
```

In dem FB_[IEDName]ServerSession-Baustein befindet sich eine Statemachine, von der die grundlegenden Zustände in jedem Server-Beispiel genutzt werden (siehe Quellcode unten). Der Server-Session-Baustein ist verantwortlich für den Verbindungsaufbau und Datenaustausch zu einem einzelnen Client. Bei mehreren gleichzeitigen Clientverbindungen werden mehrere Instanzen von diesem Baustein benötigt und instanziiert.

State 0 (Init state): Die Statemachine befindet sich in diesem Zustand, sobald das SPS-Programm gestartet wurde. Hier (und im State 1) werden Befehle zur Verwaltung einer -Server—Client-Verbindung behandelt. Im Regelfall verbleibt der Server-Session-Baustein in diesem Zustand und wartet auf einen Verbindungswunsch eines Clients. Der Server baut dann eine Verbindung auf, sobald der Verbindungswunsch erkannt wurde. Dies geschieht automatisch und benötigt keine weiteren Befehle seitens der SPS-Applikation. Falls der Server-Verbindung zum Client hergestellt hat, wird die Statemachine in den Zustand der Datenübertragung wechseln (State 10).

Es ist außerdem möglich, eine bereits hergestellte Verbindung zu schließen. Gesteuert wird dies über zwei boolesche Variablen. Diese Variablen, wenn sie gesetzt sind, aktivieren die entsprechenden Befehle (in diesem Fall sind es einmalige Methodenaufrufe am Server-Session-Baustein).

- **_bAbort:** Ruft die Methode „AbortReq“ auf, welche den Befehl zum Abbruch der Server-Verbindung zum Client aktiviert.
- **_bDisconnect:** Ruft die Methode „ReleaseReq“ auf, welche den Befehl zum geregelten Abbau einer bestehenden Server-Verbindung zum Client aktiviert. Im Regelfall ist es aber der Client, der seine aktive Verbindung zum Server beenden sollte.

Die zuvor aufgezählten Methoden benötigen für die Abarbeitung länger als einen SPS-Zyklus. Aus diesem Grund wechselt die Statemachine in einen Warte-Zustand (State 1), in dem auf die Beendigung des aktivierten Befehls gewartet wird.

State 1 (Wait State): In diesem Zustand verweilt der Server-Session-Baustein, bis die Befehlsabarbeitung zur Verwaltung der Server-Client-Verbindung nicht mehr beschäftigt (busy) ist. Solange die Verbindung abgebaut oder abgebrochen wird, befindet sich die Statemachine im State 1. Sobald der Befehl erfolgreich abgearbeitet wurde, wird die Statemachine zurück in den State 0 (Init State) versetzt.

State 10 (Data exchange): Wenn die Statemachine in diesem Zustand ist, dann ist die Server-Verbindung zum Client bereits erfolgreich hergestellt worden. Der Server-Session-Baustein ist bereit für die Datenübertragung zum Server. Die Server-Datenübertragung zum Client geschieht automatisch im Hintergrund und muss nicht seitens der SPS-Applikation initiiert werden. Es ist der Client, der durch einen Request eine Datenübertragung im Server auslöst.

Bei einer aktiven Verbindung wechselt die Statemachine zwischen dem State 0 und 10. Die Statemachine wird wieder in den State 0 versetzt, um auf Änderungen im Client-Server-Verbindungsstatus zu reagieren und sie im State 0 zu behandeln.

State 100 (Error state): Sobald ein Fehler bei der Aktivierung oder Abarbeitung eines Befehls auftritt, wird die Statemachine in den State 100 versetzt. Hier wird der Fehler geloggt und die Statemachine zurück in den State 0 versetzt.

```
FUNCTION_BLOCK FB_[IEDName]ServerSession
VAR_INPUT
    fbConnection      : FB_iec61850ConnectionClass := (ipAbortInd:=THIS^, ipAssociateInd:=THIS^, ipReleaseInd:=THIS^);
END_VAR
VAR
    _bAbort           : BOOL;
```



```

    _bDisconnect      : BOOL;
    state             : BYTE;
    eState            : E_AsyncEnvironmentState;
    bBusy             : BOOL;
    bSuccess          : BOOL;
    ipResult          : I_AsyncServiceResultClass;
    sLastErrorResult  : T_MaxString;
    fbAbortReason     : FB_ServiceErrorClass := (stError:=SUCCESS_EVENT);
    sLastAbortReason  : T_MaxString;
END_VAR

fbConnection.Execute();
eState:= fbConnection.eState;

CASE state OF
  0:
    IF _bAbort THEN
      _bAbort:= FALSE;
      bSuccess:= fbConnection.AbortReq(ipReason:=fbAbortReason, ipSink:=0, ipResult=>ipResult)
    ;
      state:= SEL(bSuccess, 100, 1);
    ELSIF eState = E_AsyncEnvironmentState.Established AND _bDisconnect THEN
      _bDisconnect:= FALSE;
      bSuccess:= fbConnection.ReleaseReq(ipSink:=0, ipResult=>ipResult);
      state:= SEL(bSuccess, 100, 1);
    ELSIF eState = E_AsyncEnvironmentState.Established THEN
      state:= 10;
    END_IF
    _bDisconnect:= FALSE;
  1:
    IF ipResult <> 0 THEN
      ipResult.Execute();
      IF NOT (bBusy:=ipResult.IsBusy()) THEN
        state:= SEL(ipResult.IsCompleted(), 100, 0);
      END_IF
    END_IF
  10:
    state:= 0;
  100:
    state:= 0;
    IF ipResult <> 0 THEN
      sLastErrorResult:= ipResult.Dump();
    END_IF
END_CASE

```

GOOSE-Publisher (optional)

TwinCAT Telecontrol Configurator kann in einem Server-Projekt, während der SPS-Code Generierung, den SPS-Code für einen GOOSE-Publisher ebenfalls generieren (siehe Codebeispiel unten). Dies ist aber nur dann möglich, wenn der Anwender vorher die GOOSE-Komponenten wie z. B. GoCBs (goose control blocks) im TwinCAT Telecontrol Configurator erstellt oder aus einer SCL-Datei (z. B. ICD-Datei) importiert hat. Standardmäßig wird bei der Codegenerierung ein Baustein mit dem Namen: „FB_[IEDName]Gse“ instanziiert und der globalen Variablenliste „TcTelecontrol“ hinzugefügt. Dieser Baustein stellt die Verbindung zwischen einem Netzwerkadapter des TwinCAT Steuerrechners, dem IED-Datenmodell und der GOOSE-Konfiguration in den GoCBs her. Die GoCBs werden in dem IED-Datenmodell (normalerweise im LLN0) instanziiert. Jeder GoCB besitzt ein Baustein-Unterelement mit dem Namen: „Publisher“. Über die „Publisher“-Methodenaufrufe kann das Publishing aus dem SPS-Code gestartet („Start“) oder gestoppt werden („Stop“). Defaultmäßig wird das Publishing beim SPS-Programmstart für alle GoCBs gestartet. Dies wird durch die mit „TRUE“ initialisierte „bStart“-Variable gesteuert. Über eine steigende Flanke an der „bStop“-Variablen kann das Publishing für alle GoCBs gestoppt werden. Die über diese Methoden abgesetzten Publisher-Befehle werden sofort ausgeführt, ohne Wartezyklen oder weitere States die nötig sind, um die Befehlsabarbeitung abzuschließen. Der hier beschriebene „Publisher“ liest die Konfiguration und aktualisiert zwar den Status des GoCBs (Attribute „GoEna“ wird z. B. auf „TRUE“ oder „FALSE“ gesetzt), er nutzt aber nicht die Client-Server-Dienste wie „SetGoCBValues“ oder „GetGoCBValues“ um den Publisher zu starten oder zu stoppen. D.h. der generierte Code implementiert bereits einen Publisher der z. B. im ersten SPS-Zyklus oder aus dem SPS-Code zum beliebigen Zeitpunkt gestartet bzw. gestoppt werden kann. Die benötigten GoCB-Konfigurationseinstellungen (GoCB-Attributwerte) können über Initialisierungswerte vorgenommen werden. Die Konfiguration der GoCBs kann aber bereits in dem TwinCAT Telecontrol Configurator geschehen. Die Initialisierungswerte werden dann während der Codegenerierung automatisch generiert und zugewiesen. Wenn der Publisher gestartet wurde und die Konfiguration des GoCBs und des Netzwerkadapters eine Übereinstimmung ergibt, dann beginnt der Publisher sofort mit dem Versenden der ersten GOOSE-Frames (inklusive der Framewiederholungen). Die „Update“-Methode hat eine besondere Bedeutung. Jedes Mal wenn sie aufgerufen wird, versendet der

„Publisher“ sofort ein neues GOOSE-Frame (und automatisch neue Framewiederholungen). D.h. die Applikation kann alle relevanten GOOSE-Dataset-Daten im IED-Datenmodell zuerst setzen und dann das Versenden der geänderten Dataset-Daten mit dem „Update“-Methodenaufwurf veranlassen. Die „Execute“-Methode muss in der übrigen Zeit zyklisch aufgerufen werden. Sie ist für das Versenden der Framewiederholungen und für die Aktualisierung der Statusinformation im GoCB verantwortlich.

```

FUNCTION_BLOCK FB_[IEDName]Gse IMPLEMENTS I_GseLinkStatusEventSink
VAR_INPUT
    fbAdapter    : FB_GseAdapterClass := (ipLinkStatus:=THIS^);
END_VAR
VAR
    eLinkStatus : E_GseLinkStatus;
    bSuccess    : BOOL;
    ipError     : I_ServiceErrorClass;
    bStart      : BOOL := TRUE;
    bStop       : BOOL;
    bUpdate     : BOOL;
END_VAR

bSuccess:= fbAdapter.Execute(ipError=>ipError);

IF bStart THEN
    bStart:= FALSE;
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb01.Publisher.Start(ipAdapter:=fbAdapter, ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb02.Publisher.Start(ipAdapter:=fbAdapter, ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb03.Publisher.Start(ipAdapter:=fbAdapter, ipError=>ipError);
ELSIF bStop THEN
    bStop:= FALSE;
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb01.Publisher.Stop(ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb02.Publisher.Stop(ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb03.Publisher.Stop(ipError=>ipError);
ELSIF bUpdate THEN
    bUpdate:= FALSE;
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb01.Publisher.Update(ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb02.Publisher.Update(ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb03.Publisher.Update(ipError=>ipError);
ELSE
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb01.Publisher.Execute(ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb02.Publisher.Execute(ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb03.Publisher.Execute(ipError=>ipError);
END_IF

```

Der GSE-Baustein implementiert die „I_GseLinkStatusEventSink“-Schnittstelle. Die Methode: „OnLinkStatusChange“ gehört zu dieser Schnittstellenimplementierung und wird immer dann aufgerufen, wenn sich der Status der Netzwerkverbindung (am Netzwerkadapter) ändert. Die SPS-Applikation kann z. B. den Netzwerkverbindungsstatus via „eLinkStatus“-Variable abfragen oder überprüfen.

```

METHOD OnLinkStatusChange
VAR_INPUT
    ipAdapter : I_GseAdapterClass;
    eStatus   : E_GseLinkStatus;
END_VAR
VAR
END_VAR
eLinkStatus:= eStatus;

```

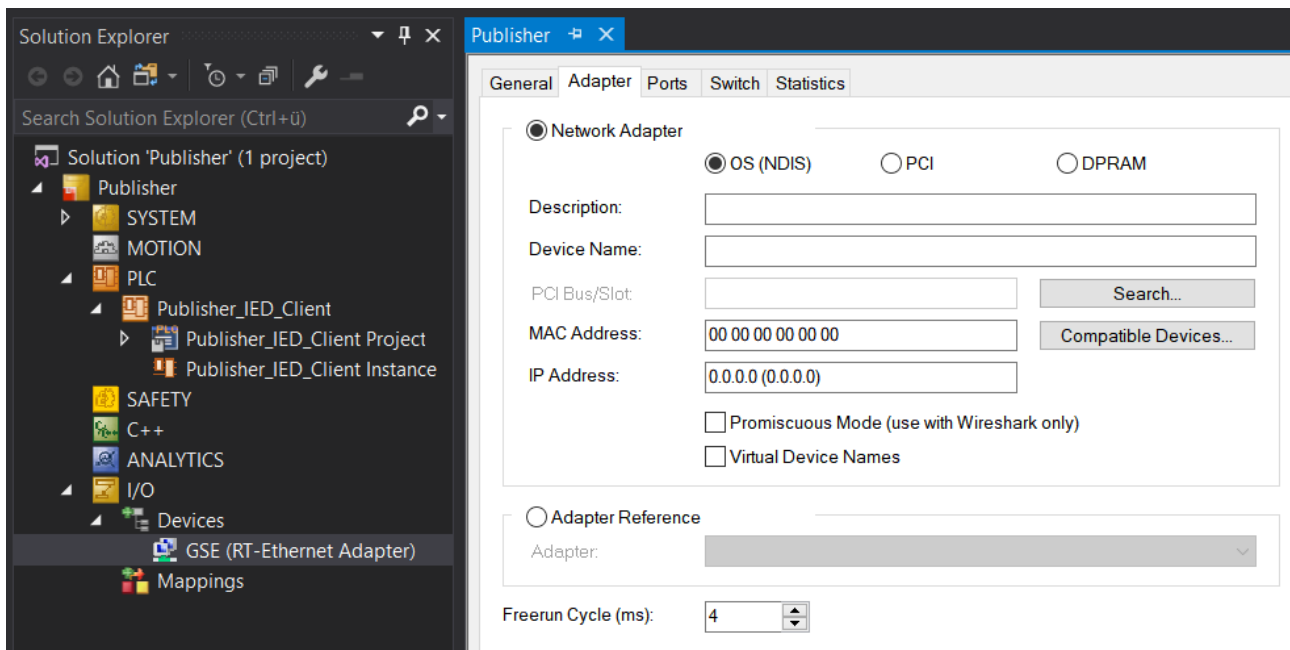
Im Projektbaum unter dem Zweig I/O-Device finden Sie eine Netzwerkadapterinstanz mit dem Namen „GSE (RT-Ethernet Adapter)“. Diese Adapterinstanz muss entsprechend konfiguriert werden, d.h. die I/O-Konfiguration muss an die vorhandene Hardware und auf die Zielplattform, auf der das Projekt laufen soll, angepasst werden.

Eine erneute I/O-Konfiguration ist auch dann nötig, wenn Sie die Zielplattform wechseln. Diese Konfiguration muss manuell in TwinCAT XAE vorgenommen werden. Zusätzlich zu der I/O-Konfiguration des Netzwerkadapters muss ein Link zwischen dem Netzwerkadapter und den SPS-Bausteinen für die Goose-Kommunikation hergestellt werden. Mit dem Link können die vom Netzwerkadapter empfangenen Daten an die Instanz des Funktionsbausteins: „FB_[IEDName]Gse“ weitergeleitet werden. In umgekehrter Richtung kann die Instanz des Funktionsbausteins „FB_[IEDName]Gse“ die zu sendenden Daten an den Netzwerkadapter weiterleiten.

Hier finden Sie weitere Informationen: [RT – Ethernet Adapter Konfiguration \[► 37\]](#).

4.6 RT - Ethernet Adapter Konfiguration

Der TwinCAT Telecontrol Configurator generiert den SPS-Standardcode für den „Publisher“ oder „Subscriber“ immer dann, wenn entsprechende Goose-Steuerblockinstanzen konfiguriert oder aus einer ICD-Datei importiert wurden. Damit diese Instanzen mit der Außenwelt kommunizieren können wird noch einen Netzwerkadapter (RT-Ethernet Adapter) benötigt, der die Goose-Nachrichten versenden (Publisher) oder empfangen kann (Subscriber). Der TwinCAT Telecontrol Configurator erstellt während der Projektgenerierung automatisch eine solche Adapterinstanz des Typs „Real-Time Ethernet Adapter (Multiple Protocol Handler)“. Sie befindet sich später im Projektbaum unter dem Zweig I/O-Devices. Diese Adapterinstanz muss entsprechend konfiguriert werden, d.h. die I/O-Konfiguration muss an die vorhandene Hardware und auf die Zielplattform, auf der das Projekt laufen soll, angepasst werden. Eine erneute I/O-Konfiguration ist auch notwendig, wenn Sie die Zielplattform wechseln. Diese Konfiguration muss manuell in TwinCAT XAE vorgenommen werden. Das nachfolgende Bild zeigt eine vom TwinCAT Telecontrol Configurator angelegte, noch nicht konfigurierte Netzwerkadapterinstanz. Um den Echtzeit-Netzwerkadapter und somit die GOOSE-Kommunikation nutzen zu können, muss das verwendete Gerät über einen kompatiblen Netzwerkadapter verfügen. Die Liste unterstützter Netzwerk-Chips finden Sie hier: [Unterstützte Netzwerk-Controller](#).

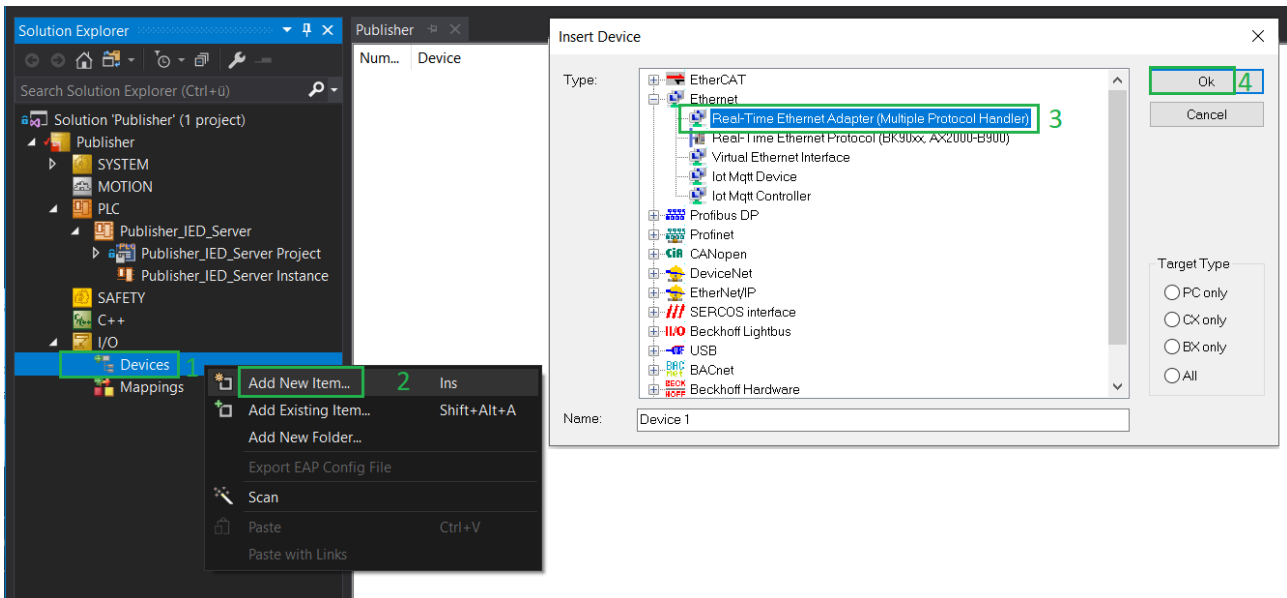


Zusätzlich zu der I/O-Konfiguration des Netzwerkadapters muss noch ein Link zwischen dem Netzwerkadapter und den SPS-Bausteinen für die Goose-Kommunikation hergestellt werden. Mit diesem Link können die vom Netzwerkadapter empfangenen Daten an die Instanz des Funktionsbausteins „FB_[IEDName]Gse“ weitergeleitet werden. In umgekehrter Richtung kann die Instanz des Funktionsbausteins „FB_[IEDName]Gse“ die zu sendenden Daten an den Netzwerkadapter weiterleiten.

Adapter manuell hinzufügen

Falls der Netzwerkadapter unterhalb der I/O-Konfiguration noch nicht vorhanden ist, kann er nachträglich hinzugefügt werden.

1. Klicken Sie im Projektbaum mit der rechten Maustaste auf **Devices**.
2. Wählen Sie aus dem Kontextmenü **Add New Item....**
3. Aus der Geräteliste **Ethernet** wählen Sie **Real-Time Ethernet Adapter (Multiple Protocol Handler)** aus.
4. Bestätigen Sie die Auswahl mit **Ok**.

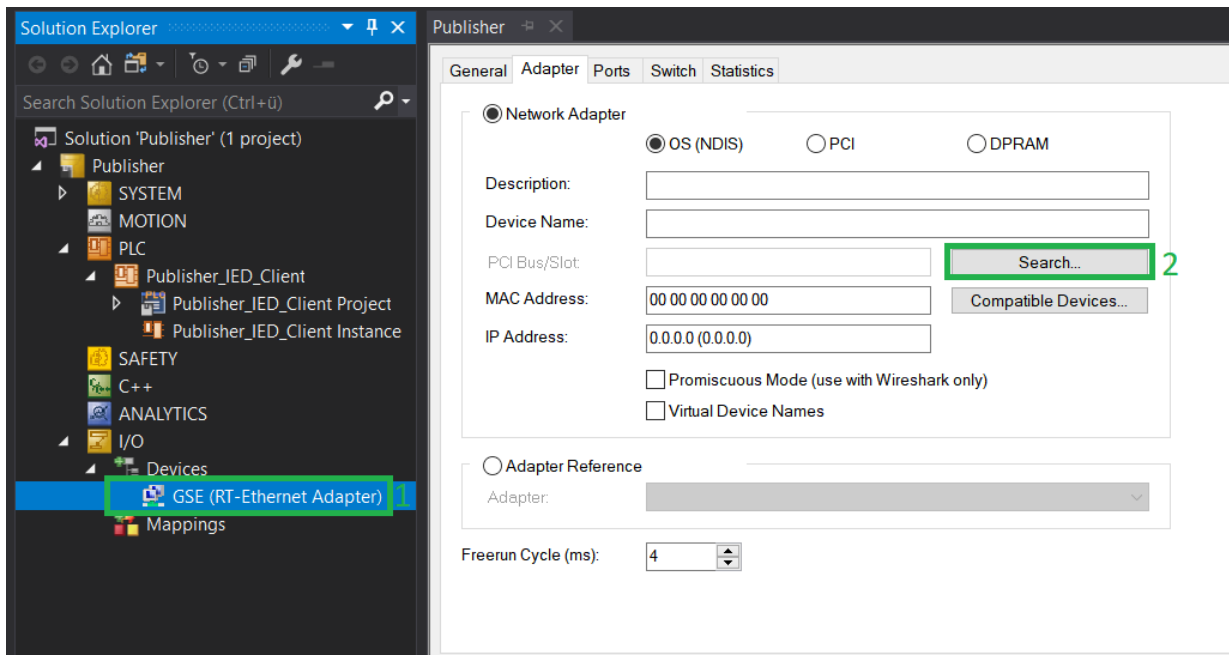


⇒ Damit ist der Adapter hinzugefügt.

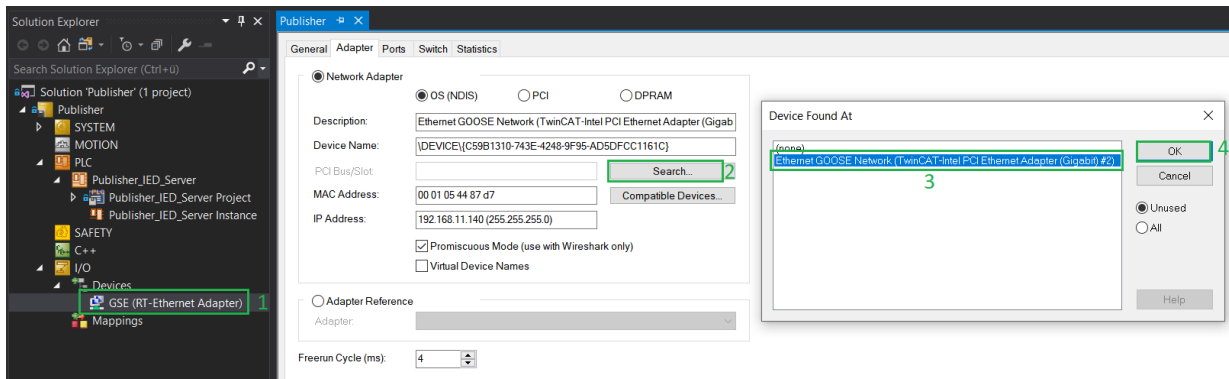
Adapter konfigurieren

✓ Öffnen Sie den Dialog mit den Adaptereigenschaften mit einem doppelten rechten Mausklick auf den Adapter.

1. Wechseln Sie zu dem Reiter **Adapter**.
2. Wählen Sie dort den Befehl **Search...** aus.



3. Wählen Sie aus der Liste der gefundenen Adapter einen Adapter aus und bestätigen dies mit **OK**.



⇒ Damit ist die Adapterkonfiguration abgeschlossen. Im nächsten Schritt muss ein Link zwischen dem Adapter und der SPS-Bausteininstanz „FB_[IEDName]Gse“ hergestellt werden.

Adapter mit der SPS-Bausteininstanz verlinken

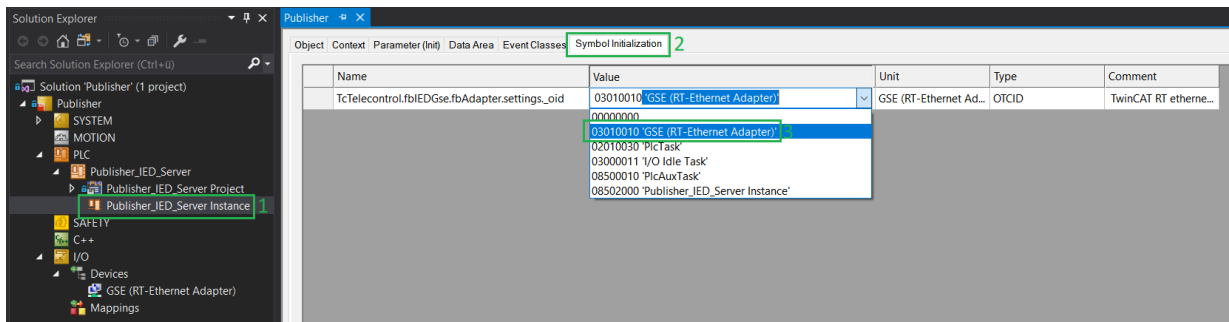
Der Link zwischen dem Netzwerkadapter und der SPS-Bausteininstanz „FB_[IEDName]Gse“ wird über eine sogenannte Objekt-ID (oid) hergestellt. Die im Projekt verfügbaren Objekt-ID's werden auf dem Dialog der SPS-Projektinstanz unter dem Reiter **Symbol Initialization** angezeigt. Das Projekt muss vorher mindestens einmal übersetzt worden sein, damit der Reiter und die Objekt-IDs erscheinen.

HINWEIS

Ein Real-Time-Netzwerkadapter kann mit max. 20 Instanzen der SPS-Bausteininstanz „FB_[IEDName]Gse“ über die Objekt-ID verlinkt werden.

✓ Öffnen Sie den Dialog mit den Adaptereigenschaften mit einem doppelten rechten Mausklick auf den Adapter.

1. Wechseln Sie zum Reiter **Symbol Initialization**.
2. Wählen Sie in der Spalte **Value** aus der Drop-Down-Box Ihren Netzwerkadapter **GSE (RT-Ethernet Adapter)** aus.
3. Speichern Sie alle Änderungen in der Solution.



⇒ Damit ist die Konfiguration und die Verlinkung des Real-Time-Netzwerkadapters abgeschlossen.

4.7 TCP Keep-Alive Messages

Ein Keep-Alive Telegramm ist eine Bestätigungsnachricht bzw. ein Acknowledge. Damit kann im Hintergrund überprüft werden, ob ein Kommunikationspartner, welcher eine Verbindung hergestellt hat, noch aktiv ist und somit noch an der Kommunikation teilnimmt. Sollte ein Kommunikationspartner nicht mehr aktiv sein, wird der Kommunikationskanal sauber und regulär geschlossen, um für einen neuen Teilnehmer frei zu sein.

Keep-Alive unter Windows

Keep-Alive kann unter Windows durch die Schlüssel „KeepAliveTime“ und „KeepAliveInterval“ in der Registrierung konfiguriert werden. Der Defaultwert für die Keep-Alive-Time ist auf zwei Stunden eingestellt, die Interval-Time, also die Zeit bis zur Wiederholung einer nicht beantworteten Alive-Anfrage, liegt Default bei einer Sekunde. **Alle Werte werden in Millisekunden angegeben**. Wenn die entsprechenden Schlüssel noch nicht in der Registrierung vorhanden sind, müssen sie wie folgt angelegt werden.

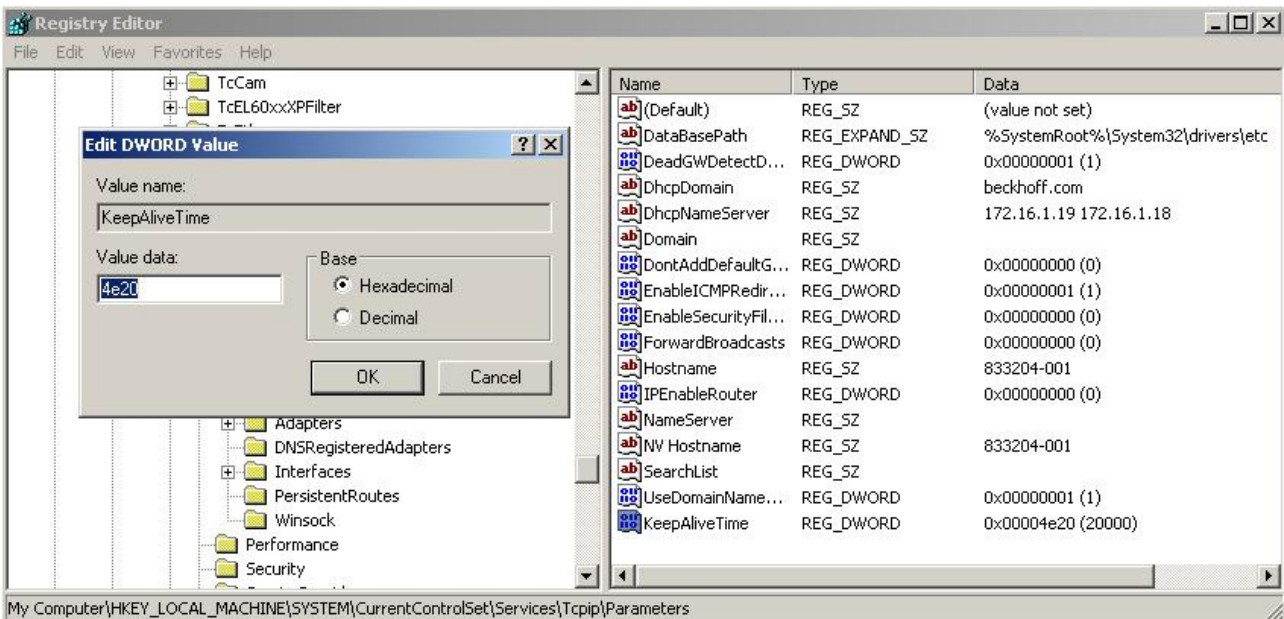
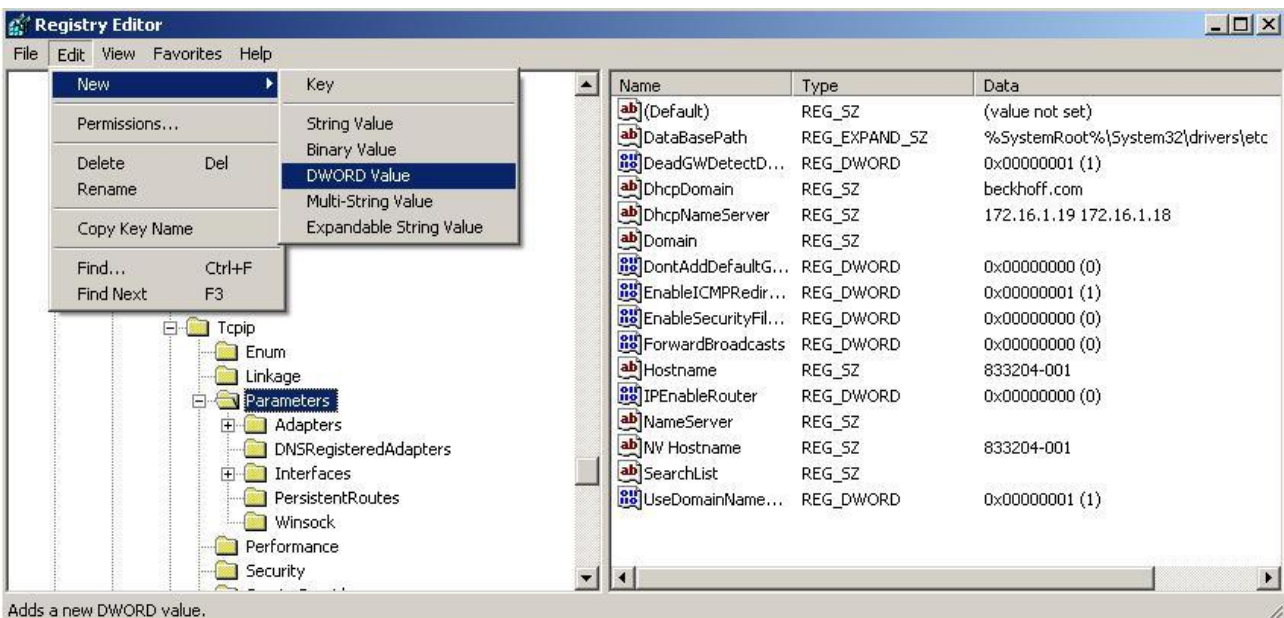
i Es handelt sich nicht um TwinCAT Einstellungen, sondern um spezifische Einstellungen des Betriebssystems. Weiterführende Informationen finden Sie daher in der Betriebssystemdokumentation z. B. auf der Microsoft Internetseite.

Unter Windows W2K, XP, Windows Embedded Standard:

- Im Ordner *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters* muss ein Objekt mit dem Namen *KeepAliveTime* als *DWORD* angelegt werden. Fügen Sie dazu unter **Edit > New > DWORD Value** ein. Stellen Sie hier als Wert *0x4E20* ein, entspricht dies 20 Sekunden.
- Im Ordner *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters* muss ein Objekt mit dem Namen *KeepAliveInterval* als *DWORD* angelegt werden. Fügen Sie dazu unter **Edit > New > DWORD Value** ein. Stellen Sie hier als Wert *0x1388* ein, entspricht dies 5 Sekunden.

Unter Windows CE:

- Im Ordner *HKEY_LOCAL_MACHINE\Comm\Tcplp\Parms* muss ein Objekt mit dem Namen *KeepAliveTime* als *DWORD* angelegt werden. Fügen Sie dazu unter **Edit > New > DWORD Value** ein. Stellen Sie hier als Wert *0x4E20* ein, entspricht dies 20 Sekunden.
- Im Ordner *HKEY_LOCAL_MACHINE\Comm\Tcplp\Parms* muss ein Objekt mit dem Namen *KeepAliveInterval* als *DWORD* angelegt werden. Fügen Sie dazu unter **Edit > New > DWORD Value** ein. Stellen Sie hier als Wert *0x1388* ein, entspricht dies 5 Sekunden.



Keep-Alive unter TC/BSD

Es werden die "sysctl" Parameter vom Betriebssystem verwendet, um TCP-Keep-Alive-Messages zu konfigurieren.

- `keepidle`: Amount of time, in milliseconds, that the connection must be idle before keepalive probes (if enabled) are sent. The default is 7200000 msec (2 hours).
- `keepintvl`: The interval, in milliseconds, between keepalive probes sent to remote machines, when no response is received on a `keepidle` probe. The default is 75000 msec.

```
> sysctl -d net.inet.tcp | grep -e keep
net.inet.tcp.keepidle: time before keepalive probes begin
net.inet.tcp.keepintvl: time between keepalive probes
net.inet.tcp.keepinit: time to establish connection
net.inet.tcp.keepcnt: Number of keepalive probes to send
net.inet.tcp.always_keepalive: Assume SO_KEEPAVIVE on all TCP connections
```

Beispiel:

`keepidle` wird auf 4000 (ms) und `keepintvl` auf 4000 (ms) gesetzt.

```
sysctl net.inet.tcp.keepidle=4000
sysctl net.inet.tcp.keepintvl=4000
```

Eine dauerhafte (persistente) Konfiguration der TCP-Keep-Alive-Meldungen wird in der Datei `/etc/sysctl.conf` vorgenommen.

Siehe auch hier: <https://www.freebsd.org/cgi/man.cgi?query=tcp>

4.8 Firewall Einstellungen

Die IEC 61850 verwendet während der Client/Server-Kommunikation (MMS-Mapping) das TCP/IP als Transportprotokoll (T-Profil). Es ist daher darauf zu achten, dass der entsprechende TCP-Port bei Benutzung einer Firewall frei geschaltet wird. Die untere Tabelle listet Ports auf, die bei der Benutzung einer Firewall zu berücksichtigen sind.

Beschreibung	Typ	Protokoll	Port
Connection Oriented Transport Protocol (COTP, ISO 8073)	Protocol	TCP	102

Konfiguration unter Windows

Die Konfiguration der Windows Firewall wird über den entsprechenden Dialog in der Systemsteuerung vorgenommen. Weitere Informationen zur Konfiguration finden Sie in der Windows bzw. Firewall Dokumentation.

HINWEIS

Port freischalten

Achten Sie bei einem Embedded Controller ohne Monitoranschluss und USB darauf, dass Sie den Port für Remote Display (Windows CE) oder Remote Desktop (Windows XP / Windows Vista) in der Firewall frei schalten. Ansonsten haben Sie keine Möglichkeit mehr, den Rechner über das Netzwerk zu administrieren.

Konfiguration unter TC/BSD

- ✓ Melden Sie sich in der Konsole an.
Standardanmeldedaten:
Login: Administrator
Passwort: 1

✓ Alternativ z.B. über die Git BASH Konsole:

```
ssh Administrator@[YourIPAddress]
```

1. Als nächstes müssen Sie die Konfigurationsdatei der Firewall so modifizieren, dass der entsprechende TCP/IP-Port freigegeben wird. Folgender Befehl öffnet die Konfigurationsdatei im Editor:

```
doas ee /etc/pf.conf
```

2. Folgender Eintrag wird der Konfigurationsdatei hinzugefügt (wenn nicht bereits vorhanden):

```
#allow traffic to TCL/IP port 102  
Pass in quick proto tcp to port 102 keep state
```

3. Mit Escape wird das Menü im Editor aufgerufen.

4. Speichern Sie die Änderungen und beenden den Editor.

⇒ Konfiguration abgeschlossen.

Weitere nützliche Befehle:

Gerät rebooten: `doas shutdown -r now`

Firewall deaktivieren: `doas pfctl -d`

Firewall aktivieren: `doas pfctl -e`

5 Konfiguration

5.1 Grundlagen und Einstieg

Der TwinCAT Telecontrol Configurator ist für die grafische Konfiguration des IEC 61850 bzw. IEC 61400-25 Datenmodells gedacht. Dabei können ICD-Dateien eingelesen, erstellt und modifiziert werden. Durch eine automatische SPS-Code-Erzeugung wird die Konfiguration für die TwinCAT-Steuerung verfügbar gemacht.

Funktionen der Haupt-Toolbar

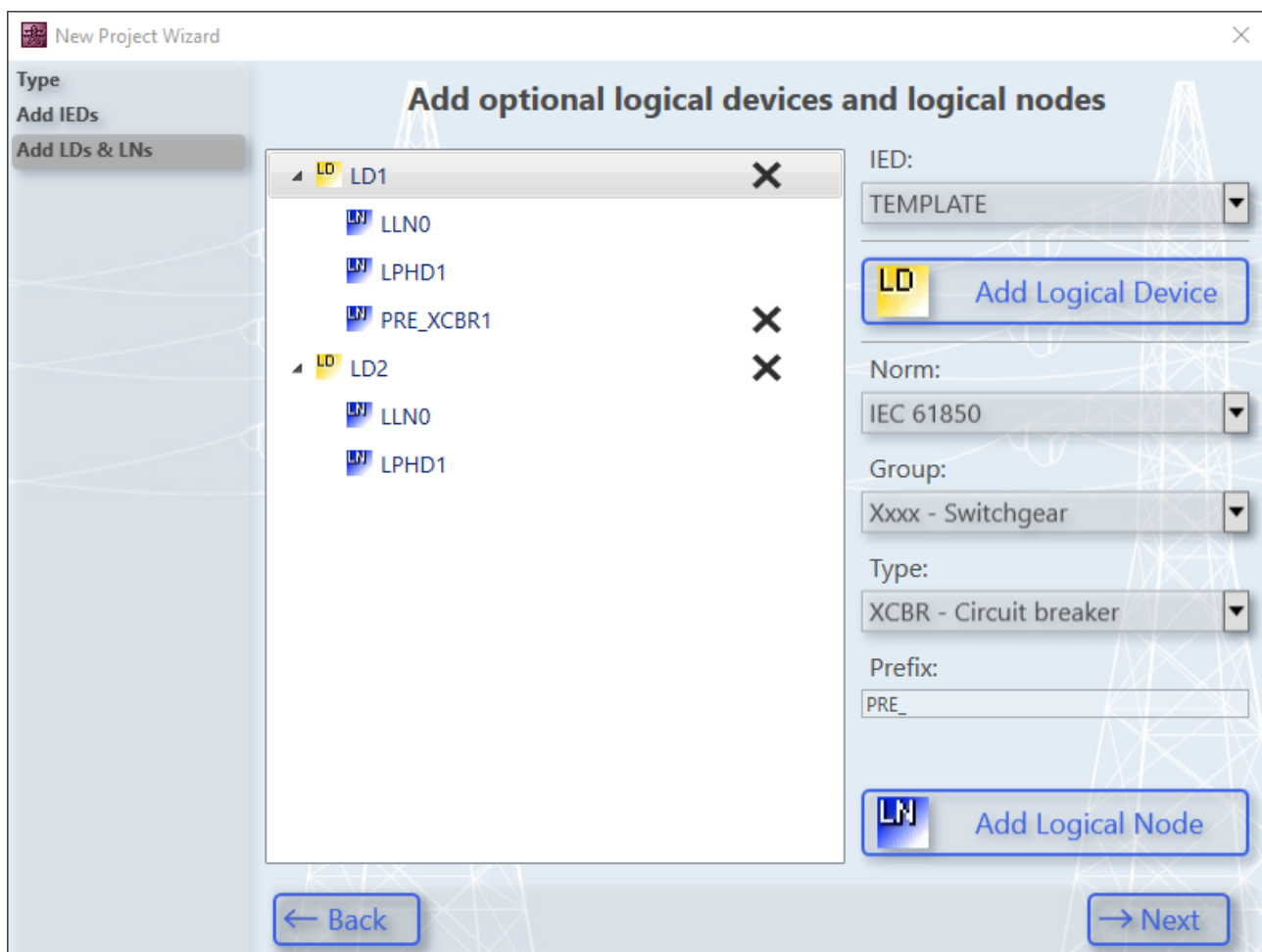


Die Buttons sind in der nachfolgenden Tabelle von links nach rechts beschrieben.

New Project	Erstellt eine neue Konfiguration, indem zunächst ein Speicherort für eine neue ICD-Datei ausgewählt werden kann. Folgend öffnet sich ein Wizard zur Vorkonfiguration des Datenmodelles.
Open ICD-file	Importiert/öffnet eine ICD-Datei
Save ICD-file	Exportiert/speichert eine aktuell geöffnete ICD-Datei
Save ICD-file as...	Exportiert/speichert eine ICD-Datei unter einem neuen Namen
Generate PLC code	Erzeugt SPS-Code für die erstellte Konfiguration

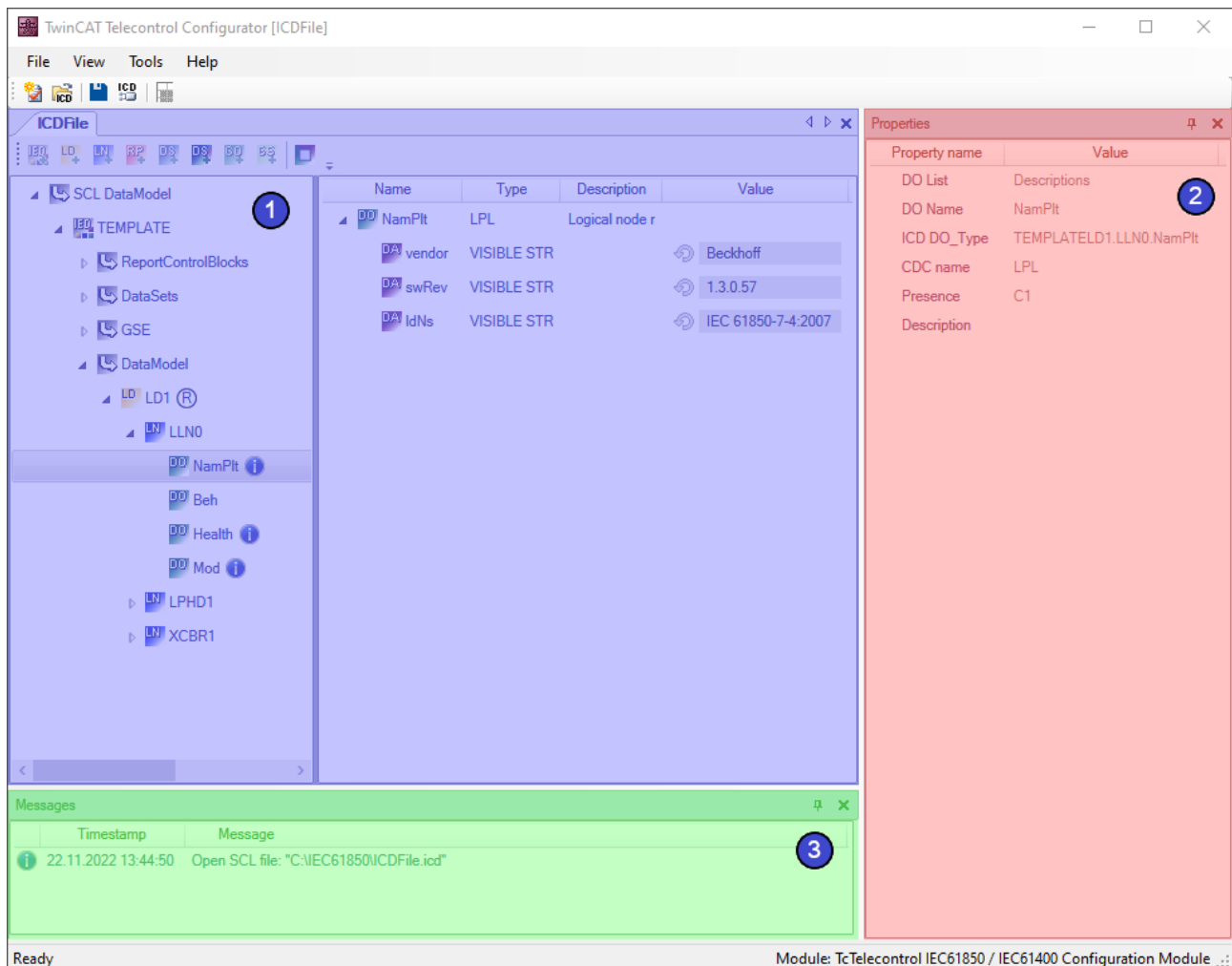
Durch die Betätigung des Buttons **New Project** in der Toolbar oder im Menü des Configurators öffnet sich zunächst ein Dialog, in dem eine neue ICD-Datei abgespeichert werden kann.

Im Folgenden öffnet sich der hier dargestellte Wizard zur Vorkonfiguration des Datenmodelles. In diesem besteht die Möglichkeit, allgemeine Einstellungen für das Projekt zu tätigen und die unterstützten Kontrollblockarten zum Datenmodell hinzuzufügen.



Das **Importieren/Öffnen** und das **Exportieren/Speichern** von ICD-Dateien sollte intuitiv bedienbar sein. Ein konfiguratorspezifisches Dateiformat gibt es nicht. Eine weitere wesentliche Funktion ist das Generieren von SPS-Code. Dies können Sie über den Button **Generate PLC code** starten. Dabei wird die gesamte Datenmodell-Konfiguration in Strukturierten Text konvertiert und grundlegende Bausteine, sowie Aufrufe mit erzeugt.

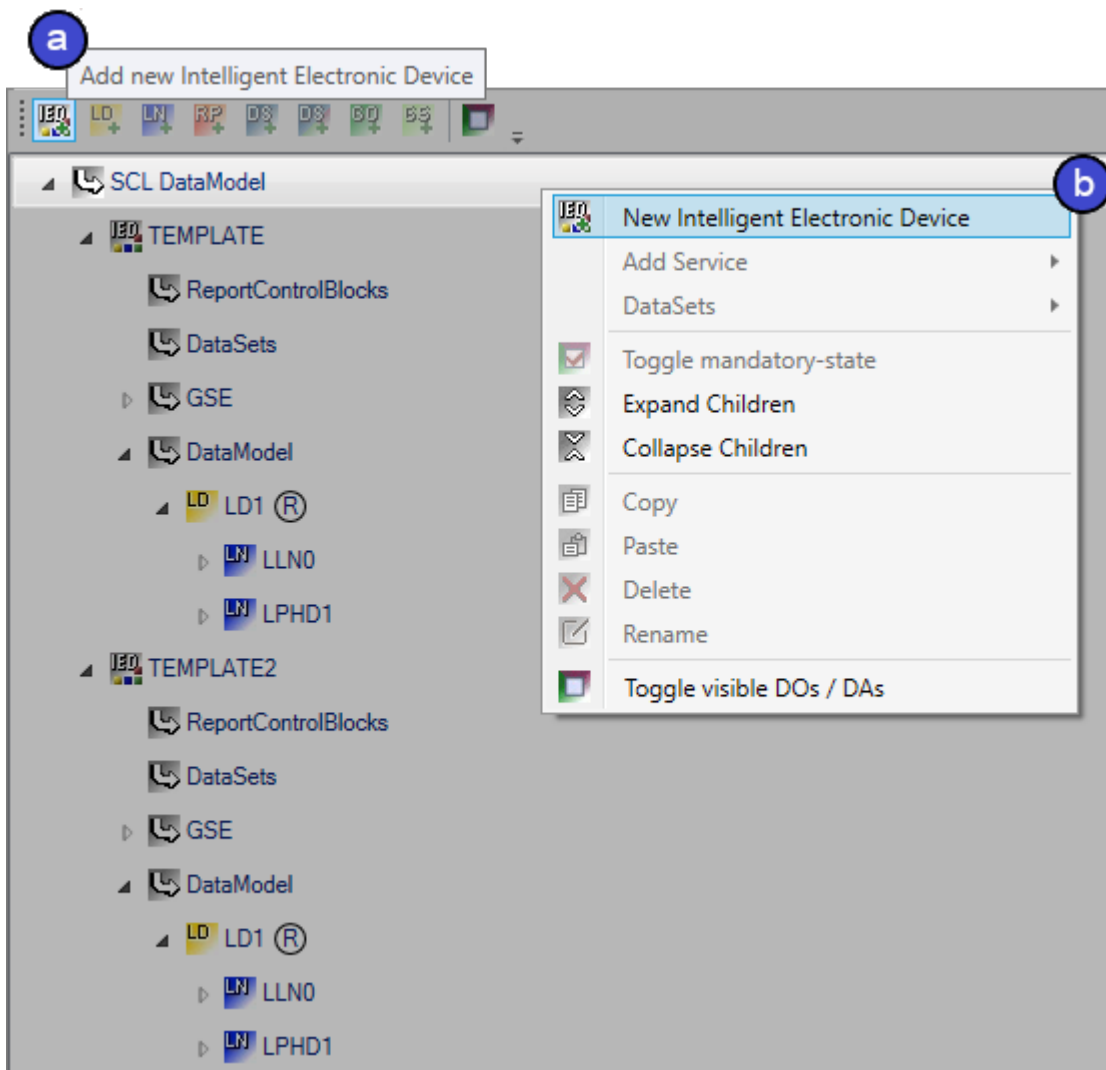
Die Grundarchitektur des Telecontrol Configurator gliedert sich in drei wesentliche Teile:



1. Der Arbeitsteil in dem das Datenmodell definiert wird. Es können unter anderem optionale Objekte hinzugefügt oder abgewählt, Datasets hinzugefügt, Reportings und GOOSE-Control-Blöcke konfiguriert, sowie Defaultwerte gesetzt werden.
2. In den Properties finden Sie weiterführende Optionen und Informationen.
3. Das Meldungenfenster gibt Informationen und Fehlermeldungen aus.

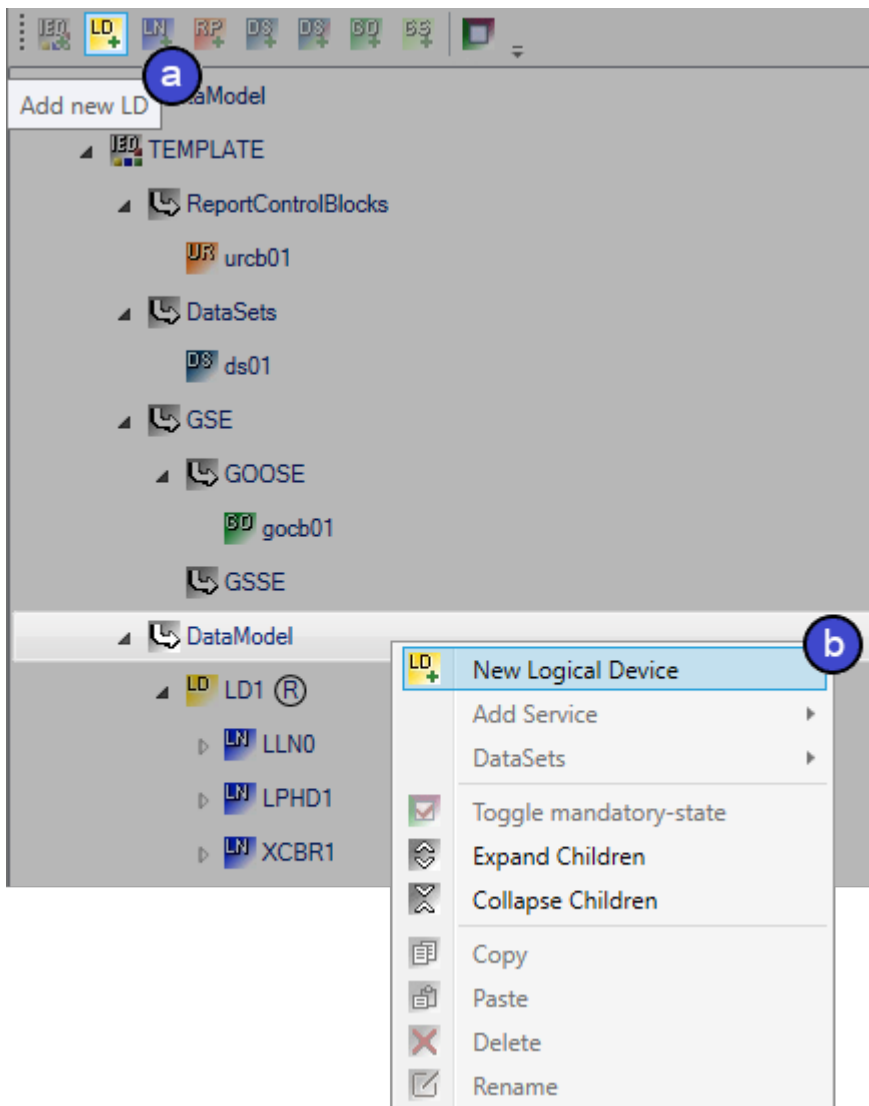
Intelligent Electronic Devices hinzufügen

Um weitere Intelligent Electric Devices hinzuzufügen, können Sie den **SCL-DataModel**-Knoten im Datenmodell anklicken, damit Sie über die Toolbar im Projektfenster (a) oder über das Kontextmenü des Knotens (b) ein **New Intelligent Electronic Device** ausführen können.



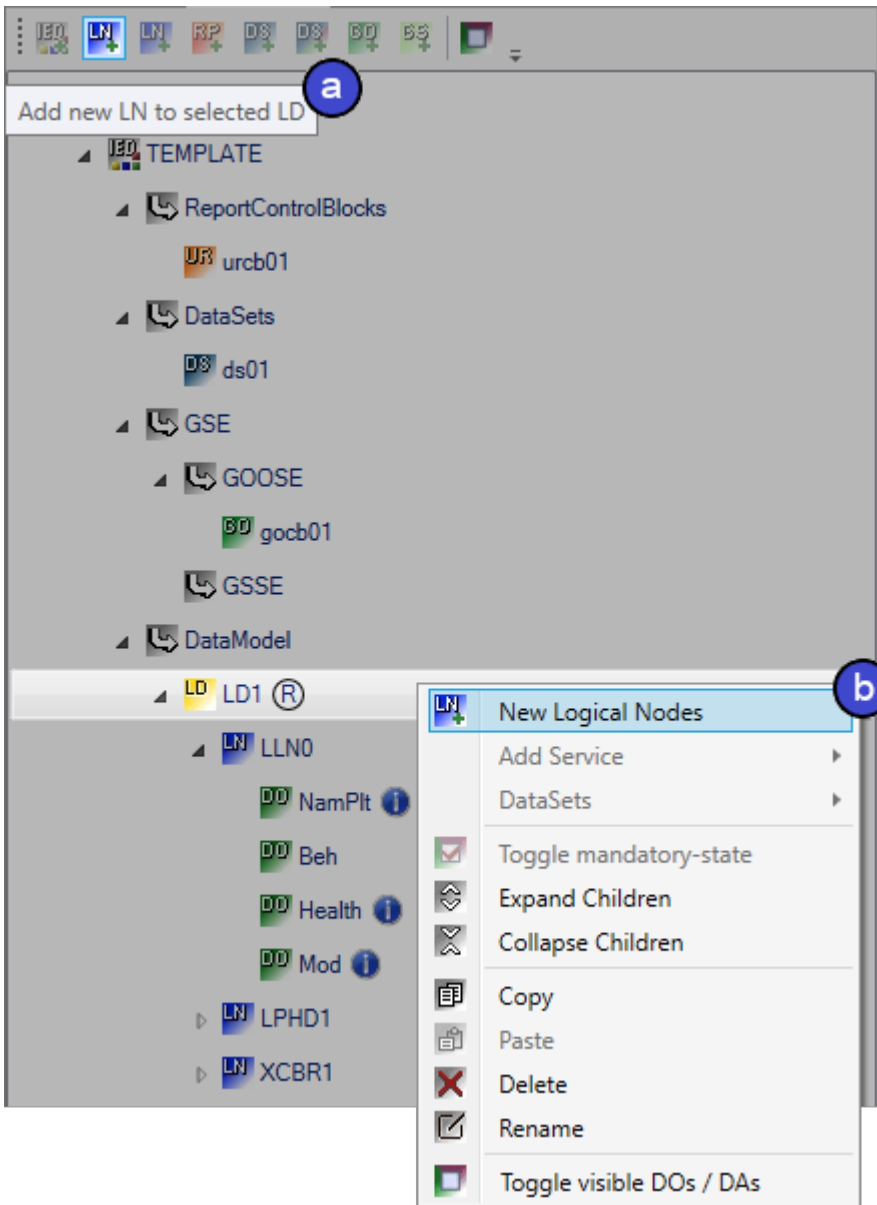
Logical Devices hinzufügen

Um weitere Logical Devices hinzuzufügen, können Sie den **DataModel**-Knoten des jeweiligen Intelligent Electronic Devices anklicken, damit Sie über die Toolbar im Projektfenster (a) oder über das Kontextmenü des Knotens (b) ein **New Logical Device** ausführen können.



Logical Nodes hinzufügen

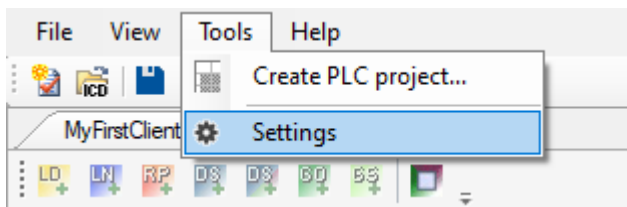
Um weitere Logical Nodes zu konfigurieren, können Sie das Device anklicken, damit Sie über die Toolbar im Projektfenster (a) oder über das Kontextmenu des Knotens (b) ein **New Logical Nodes** ausführen können.



In diesem Fall öffnet sich der nachfolgende Wizard. Dort können Sie den Standard auswählen, aus dem Sie sich bedienen möchten. Nachfolgend die Gruppe der Logical Nodes und den Typ des Knoten selbst. Ein Präfix kann optional vergeben werden.

Standardeinstellungen

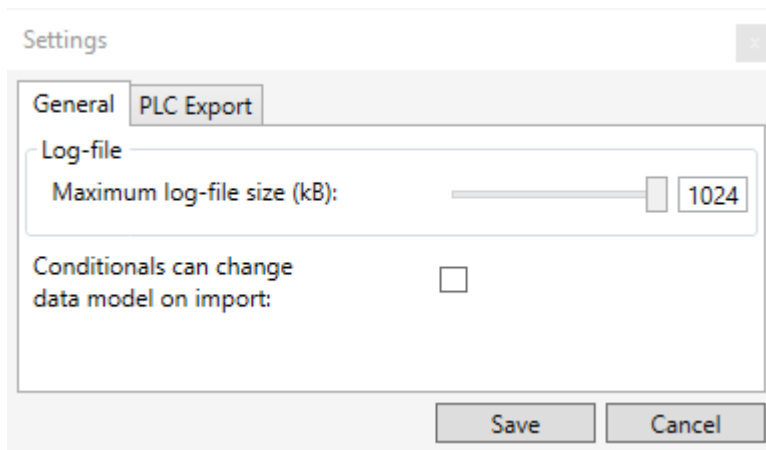
Der Telecontrol Configurator bietet zudem die Möglichkeit, einige Standardeinstellungen projektübergreifend zu speichern. Öffnen Sie die Einstellungen über das Menü **Tools** in der Haupt-Toolbar und wählen Sie die Aktion **Settings**.



Folgend sehen Sie ein neues Fenster, das sich in 2 Teilbereiche aufgliedert. Zum einen **allgemeine Konfigurator-Einstellungen** und zum anderen Standardeinstellungen für die **SPS-Code-Generierung**. Diese Standardeinstellungen dienen als Ausgangswerte und können vor der SPS-Code-Generierung angepasst werden.

In den **allgemeinen Konfigurator-Einstellungen** haben Sie die Möglichkeit, die Maximalgröße der Log-Datei zu bestimmen. Hier können Sie einen beliebigen Betrag zwischen 1 kB und 1024 kB wählen. Sollten Sie einen Wert einstellen, der kleiner als die aktuelle Größe der Log-Datei ist, so werden die ältesten Log-Einträge entfernt, bis die Dateigröße dem neuen Wert entspricht.

Zudem gibt es hier die Option, bedingungsabhängige Data Objects beim ICD-Datei-Import zu behandeln. Diese Option dient dazu, das Datenmodell normkonform zu importieren. Da dies unter Umständen ihr bestehendes Datenmodell verändern kann, ist diese Einstellung optional. Sollte diese Einstellung aktiv sein, werden Sie beim ICD-Import über automatisch durchgeführte Veränderungen des Datenmodells im Nachrichtenfenster informiert.

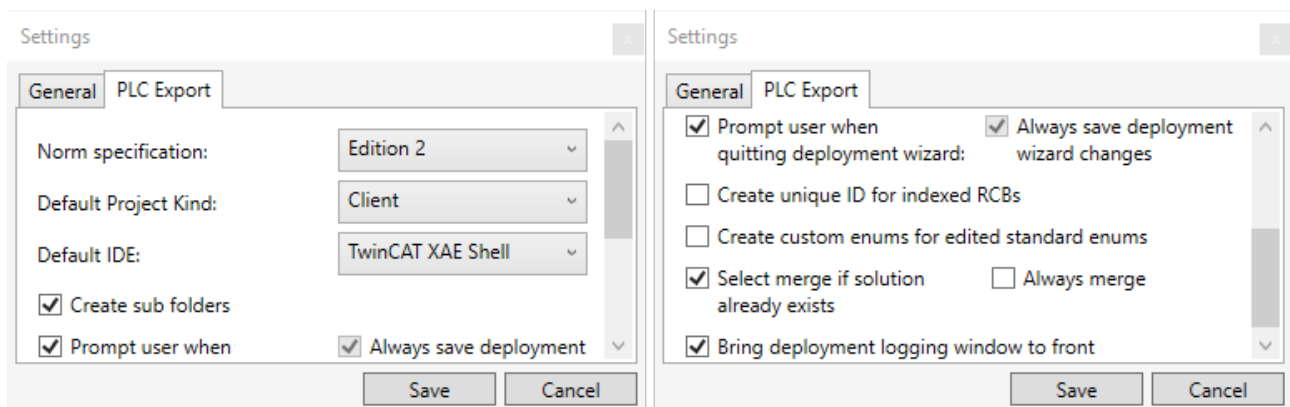


In den **Einstellungen für die SPS-Code-Generierung** können Sie Standardwerte einstellen. So können Sie festlegen, ob Sie standardmäßig einen Client oder einen Server generieren wollen oder welche Entwicklungsumgebung Sie zur Generierung nutzen möchten. Hier können Sie außerdem einstellen, ob standardmäßig Unterordner zur Organisation des Datenmodells generieren werden sollen.

Zudem können Sie einstellen, ob beim vorzeitigen Beenden des Deploy TwinCAT Telecontrol-Dialoges eine Abfrage angezeigt wird, ob die getätigten Einstellungen trotzdem bis zum nächsten Öffnen des Dialoges gespeichert bleiben sollen. Hierbei können Sie auch auswählen, dass die getätigten Einstellungen nie gespeichert bleiben sollen.

Die folgenden Optionen sind für den zu generierenden SPS-Code. „Create unique ID for indexed RCBs“ dient dazu, dass eindeutige Report IDs für indexierte Report Control Blocks generiert werden. Sollte ein normspezifisches Enum durch Bearbeitung nicht mehr mit den standardmäßig zugewiesenen Werten übereinstimmen, so kann mit der „Create custom enums for edited standard enums“-Option festgelegt werden, dass hierfür ein neues benutzerdefiniertes Enum erstellt wird, statt das vorhandene Enum aus der SPS-Bibliothek zu nutzen.

Es besteht zudem die Möglichkeit, beim Start des Deploy TwinCAT Telecontrol-Dialoges automatisch die Option „Merge to existing Project“ auszuwählen, falls ein SPS-Projekt bereits am Speicherort der jeweiligen ICD-Datei vorhanden ist. Mit der „Always merge“ Option können Sie hierbei bestimmen, ob dies automatisch bei jedem Start des Dialoges ausgewählt werden soll oder nur beim ersten Dialogstart des aktuellen ICD-Projektes. Mit der letzten Option können Sie einstellen, ob das Log-Fenster während der Code-Generierung standardmäßig im Vordergrund bleiben soll.



Log-Datei

Kritische Fehler im Telecontrol Configurator werden in einer Log-Datei gespeichert. Sie finden diese im Installationsverzeichnis des Konfigurators. Die Dateigröße der Log-Datei können Sie in den allgemeinen Konfigurator-Einstellungen bestimmen, standardmäßig ist sie auf 1024kB eingestellt. Wenn die Dateigröße den eingestellten Maximalwert überschreitet, werden die jeweils ältesten Log-Einträge aus der Datei gelöscht, um den Maximalwert einzuhalten.

Die Log-Datei ist im XML-Format aufgebaut. Ein Log-Eintrag enthält immer Informationen zum Schweregrad der Meldung und zu welchem Zeitpunkt die Meldung aufgetreten ist. Sollte der Fehler bei der SPS-Code-Generierung auftreten, so wird zudem die Kategorie PLC Export gespeichert.

Jeder Log-Eintrag beinhaltet zudem einen lesbaren Message-Eintrag zur selbstständigen Fehlerbehebung. Dieser ist vorgesehen für Fälle, in denen ein Fehler durch Fehlanwendung entsteht, beispielsweise wenn versucht wird, eine invalide ICD-Datei zu importieren. Der verschlüsselte Stack-Eintrag hilft bei der Lokalisierung des Fehlers, falls es sich um einen nicht selbstständig lösbaren Fehler am Telecontrol Configurator handelt und Sie sich damit an unseren Support wenden.

```
<Log Severity="Error" Timestamp="2021-07-14T15:38:56.1067575+02:00">
  <Message><![CDATA[Object reference not set to an instance of an object.]]></Message>
  <Stack><![CDATA[CD-58-70-38-BD-EF-BC-3B-03-06-B4-05-D1-B3-BA-97-14-BE-2B-D2-5A-BA-50-C6]]></Stack>
</Log>
<Log Category="PLC Export" Severity="Error" Timestamp="2021-07-15T11:45:08.9454274+02:00">
  <Message><![CDATA[Default Library 'Tc3_Gse, * (Beckhoff Automation GmbH)' not found! (I]]></Message>
  <Stack><![CDATA[CD-58-70-38-BD-EF-BC-3B-03-06-B4-05-D1-B3-AD-9B-24-BE-00-D2-57-B0-4C-D1]]></Stack>
</Log>
```

5.2 Menüleiste

Navigation

File View Tools Help

File:

Grundfunktionen wie Erstellen/Öffnen/Speichern/Schließen von ICD Dateien, Telecontrol Configurator Beenden.

View:

Funktionen zur Oberfläche des Telecontrol Configurator.

Tools:

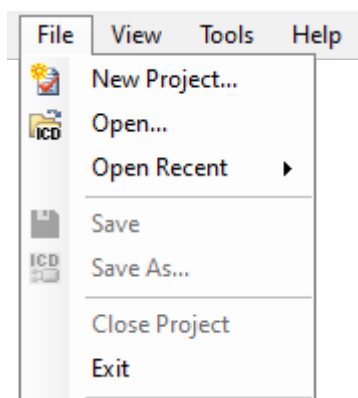
Funktion zur SPS-Code-Erzeugung und Einstellungen.

Help:

Informationen über den Telecontrol Configurator.

5.2.1 File

Grundfunktionen wie Erstellen/Öffnen/Speichern/Schließen von ICD Dateien, Telecontrol Configurator Beenden.



New Project...

Erstellt eine neue Konfiguration, indem zunächst ein Speicherort für eine neue ICD-Datei ausgewählt werden kann. Nachfolgend öffnet sich ein Wizard zur Vorkonfiguration des Datenmodelles.

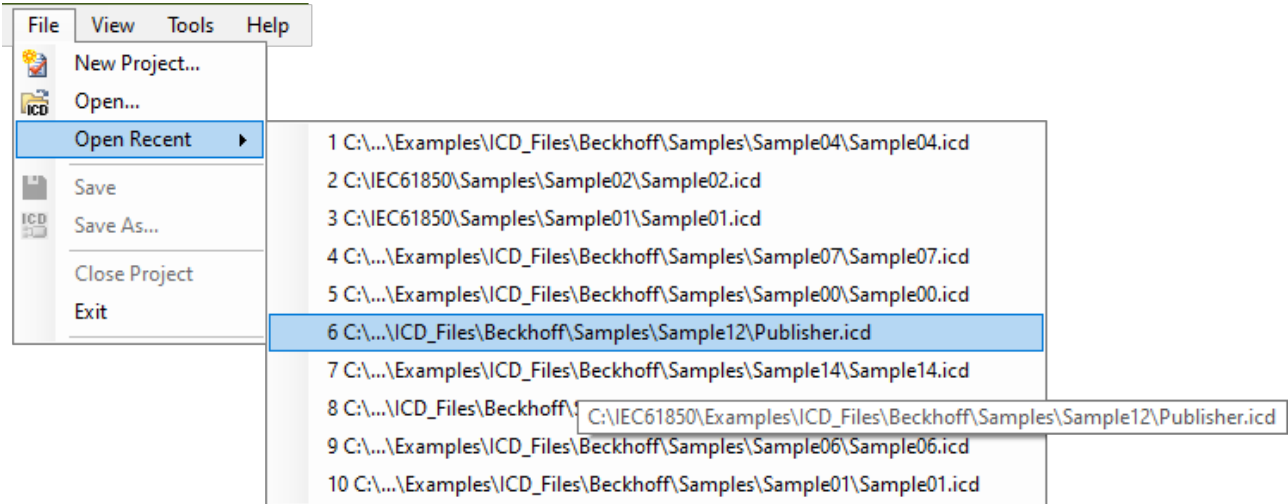
[Vollständige Beschreibung \[▶ 43\]](#)

Open...

Importiert/öffnet eine ICD-Datei.

Open Recent ▶

Importiert/öffnet eine ICD-Datei aus einer Liste der 10 zuletzt mit dem Telecontrol Configurator geöffneten Projekte.



Lange Pfade werden gekürzt dargestellt, den vollständigen Pfad können Sie im jeweiligen Tooltip sehen.

Save

Exportiert/speichert eine aktuell geöffnete ICD-Datei.

Save As...

Exportiert/speichert eine ICD-Datei unter einem neuen Namen.

Close Project

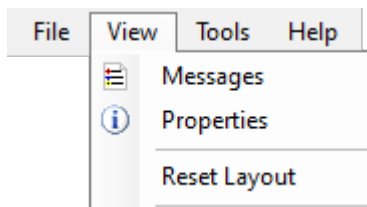
Schließt die aktuell geöffnete ICD-Datei.

Exit

Beendet den Telecontrol Configurator.

5.2.2 View

Funktionen zur Oberfläche des Telecontrol Configurator.



Messages

Zeigt das Nachrichtenfenster an.

Properties

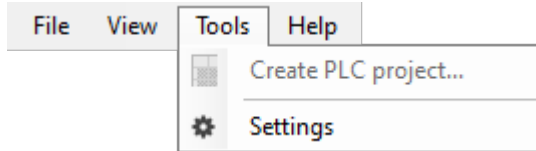
Zeigt das Eigenschaftenfenster an.

Reset Layout

Setzt das Layout des Telecontrol Configurator zurück.

5.2.3 Tools

Funktion zur SPS-Code-Erzeugung und Einstellungen.



Create PLC project...

Öffnet den Dialog zur automatischen Code-Erzeugung.

[Vollständige Beschreibung \[► 74\]](#)

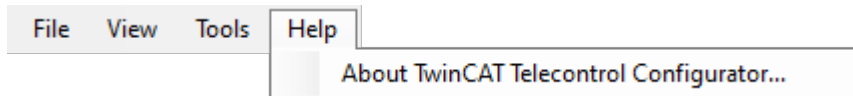
Settings

Öffnet die allgemeinen Standardeinstellungen des Telecontrol Configurator.

[Vollständige Beschreibung \[► 43\]](#)

5.2.4 Help

Informationen über den Telecontrol Configurator.



About TwinCAT Telecontrol Configurator...

Öffnet einen Dialog mit aktuellen Informationen über den Konfigurator. Hier sehen Sie unter anderem die aktuell installierte Versionsnummer und die Lizenzen der genutzten Drittanbieter (3).

Die Versionsnummern stehen für die Produktversion (1) und Konfiguratorversion (2).

About TwinCAT Telecontrol Configurator

Beckhoff TF6510 IEC61850 Telecontrol v3.1.96.0 **1**

TwinCAT.Telecontrol.IEC61850 v3.1.96.0 **2**

Copyright © Beckhoff Automation GmbH & Co. KG 2022

Beckhoff Automation GmbH & Co. KG



Included standard - DB version:

IEC 61850 - v1.0.1.0
IEC 61400-25 - v1.0.1.0

Third party licenses **3**

IEC Code Components End-user licence agreement

Code Components in IEC standards (International Standards, Technical Specifications or Technical Reports) which have been identified and approved for licensing, are licensed subject to the following conditions:

Redistributions of software must retain the Copyright Notice, this list of conditions and the

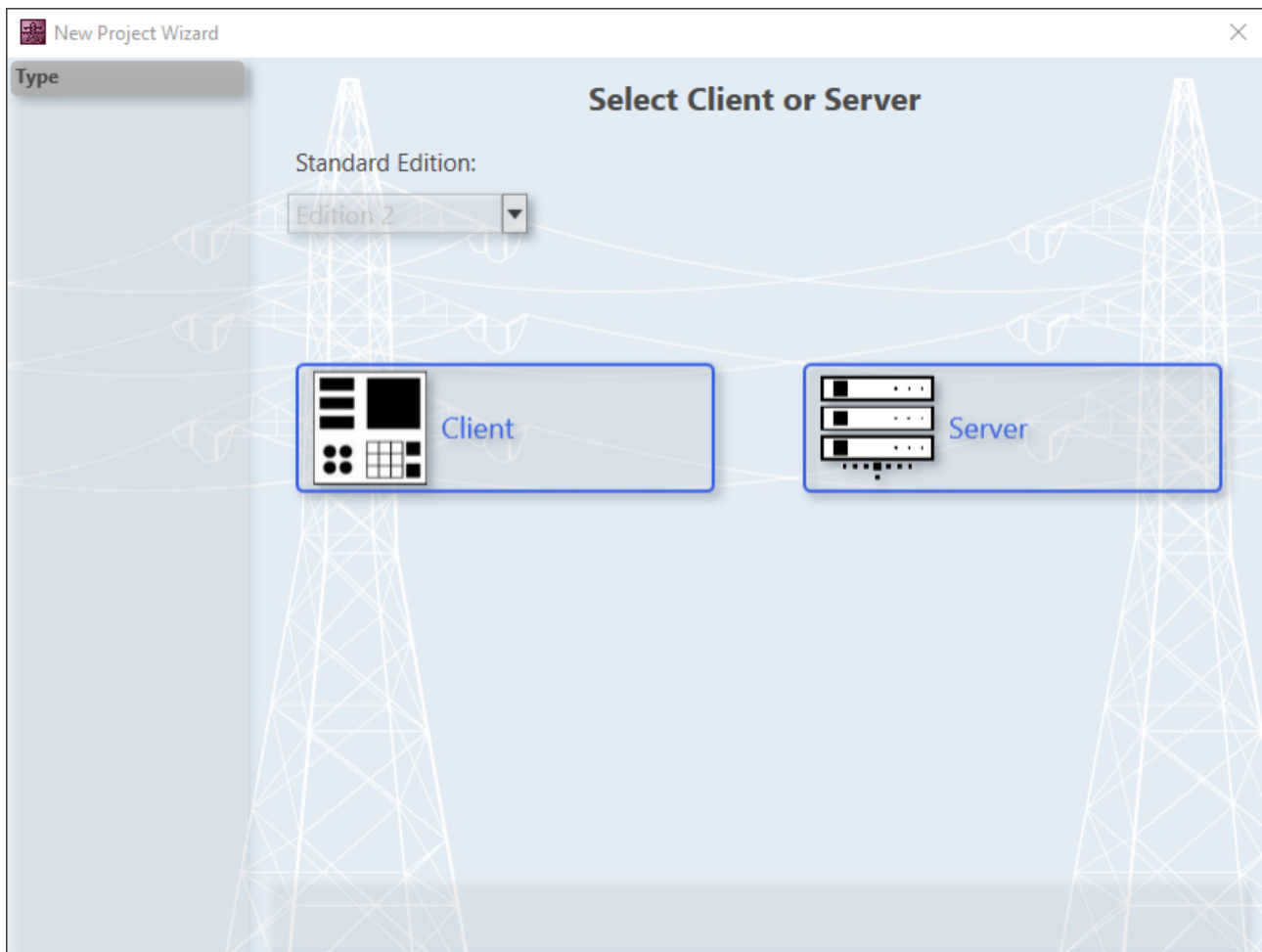
OK

5.3 New Project-Wizard

Mit Hilfe des New Project-Wizards können Sie ein neues Datenmodell vorkonfigurieren, indem Sie den einzelnen Schritten folgen. Es besteht die Möglichkeit, allgemeine Einstellungen für das Projekt zu tätigen und die unterstützten Kontrollblockarten zum Datenmodell hinzuzufügen.

Projekttyp auswählen

Zu Beginn des Wizards können Sie entscheiden, ob Sie einen Client oder Server konfigurieren möchten. Auf das Datenmodell hat dies keine Auswirkung. Diese Voreinstellung kann jederzeit in der PLC-Code-Erzeugung angepasst werden.

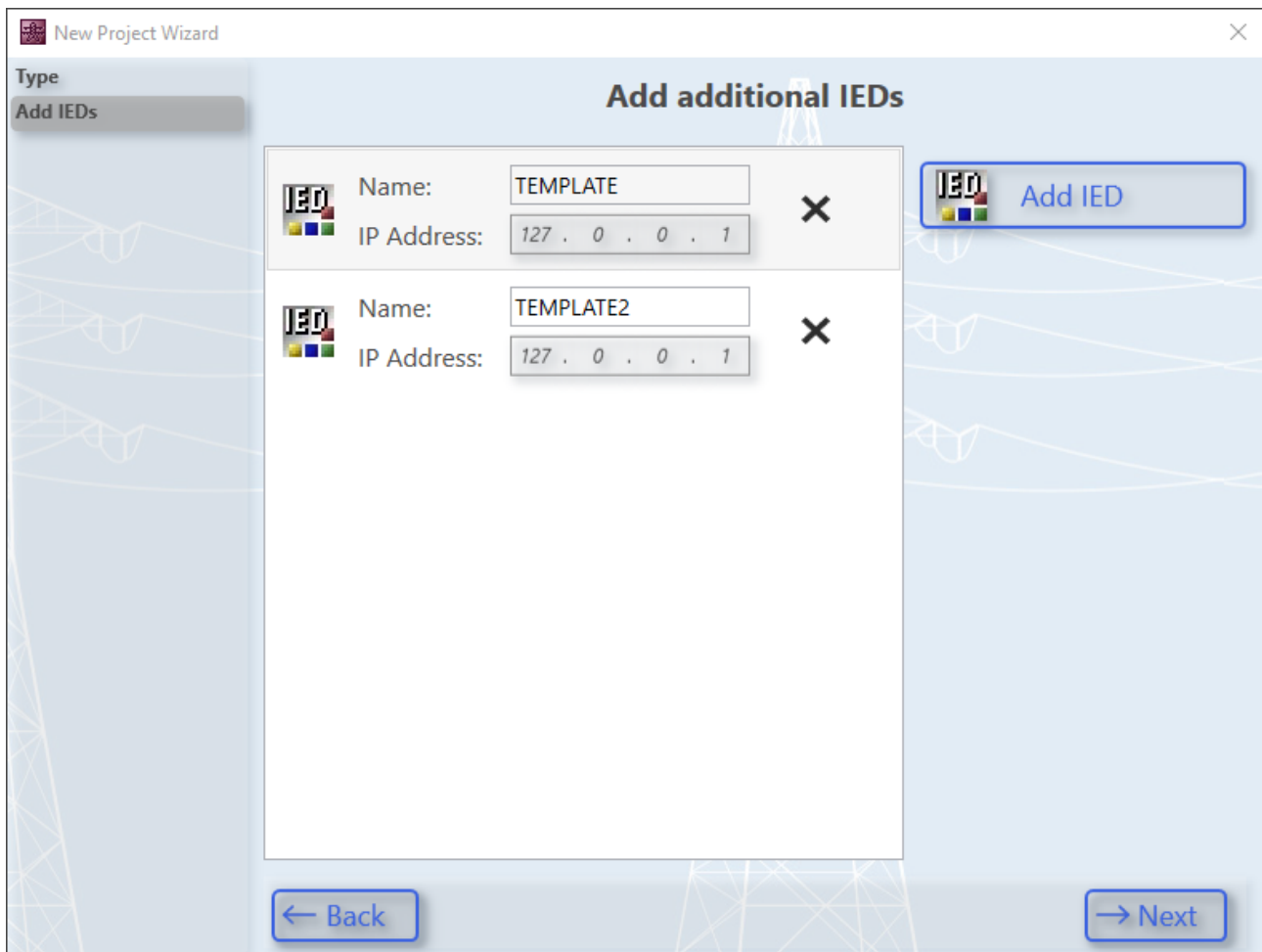


Intelligent Electronic Devices hinzufügen

Auf der folgenden Seite können Sie zusätzliche Intelligent Electronic Devices, kurz IEDs zum neuen Projekt hinzufügen. Zur Konfiguration ist mindestens ein IED notwendig.

Sie haben die Möglichkeit, jedem Intelligent Electronic Device einen Namen und eine IP-Adresse zuzuweisen. Im Falle einer Server-Konfiguration ist dies die IP-Adresse, unter der der Server erreicht werden kann. Bei der Client-Konfiguration die IP-Adresse des zu verbindenden Servers.

Das Datenmodell ist standardmäßig mit einem Intelligent Electronic Device vorkonfiguriert.

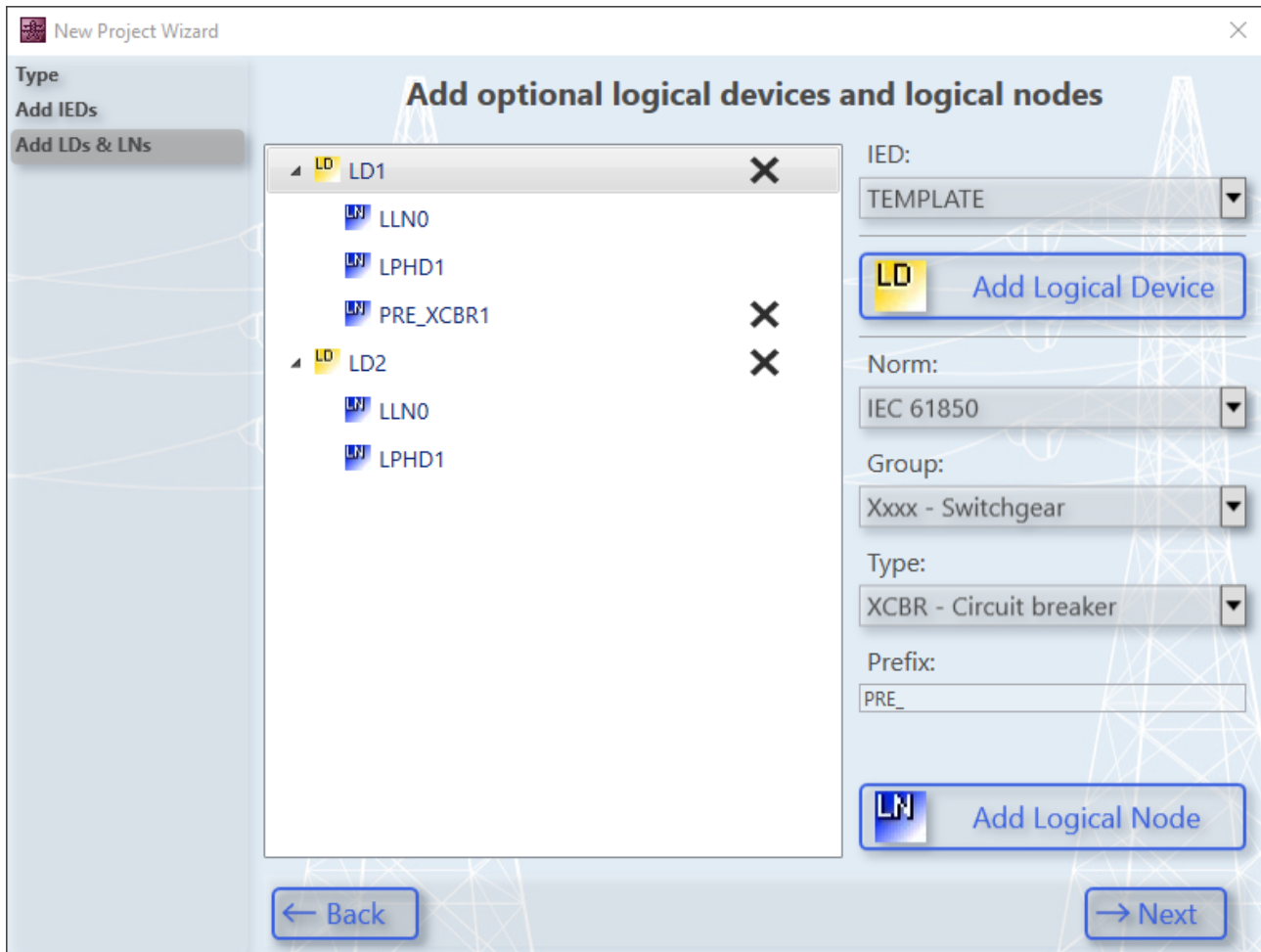


Logical Nodes hinzufügen

Auf der folgenden Seite können Sie Logical Devices zum ausgewählten Intelligent Electronic Device hinzufügen. Zur Konfiguration ist mindestens ein Logical Device pro Intelligent Electronic Device notwendig.

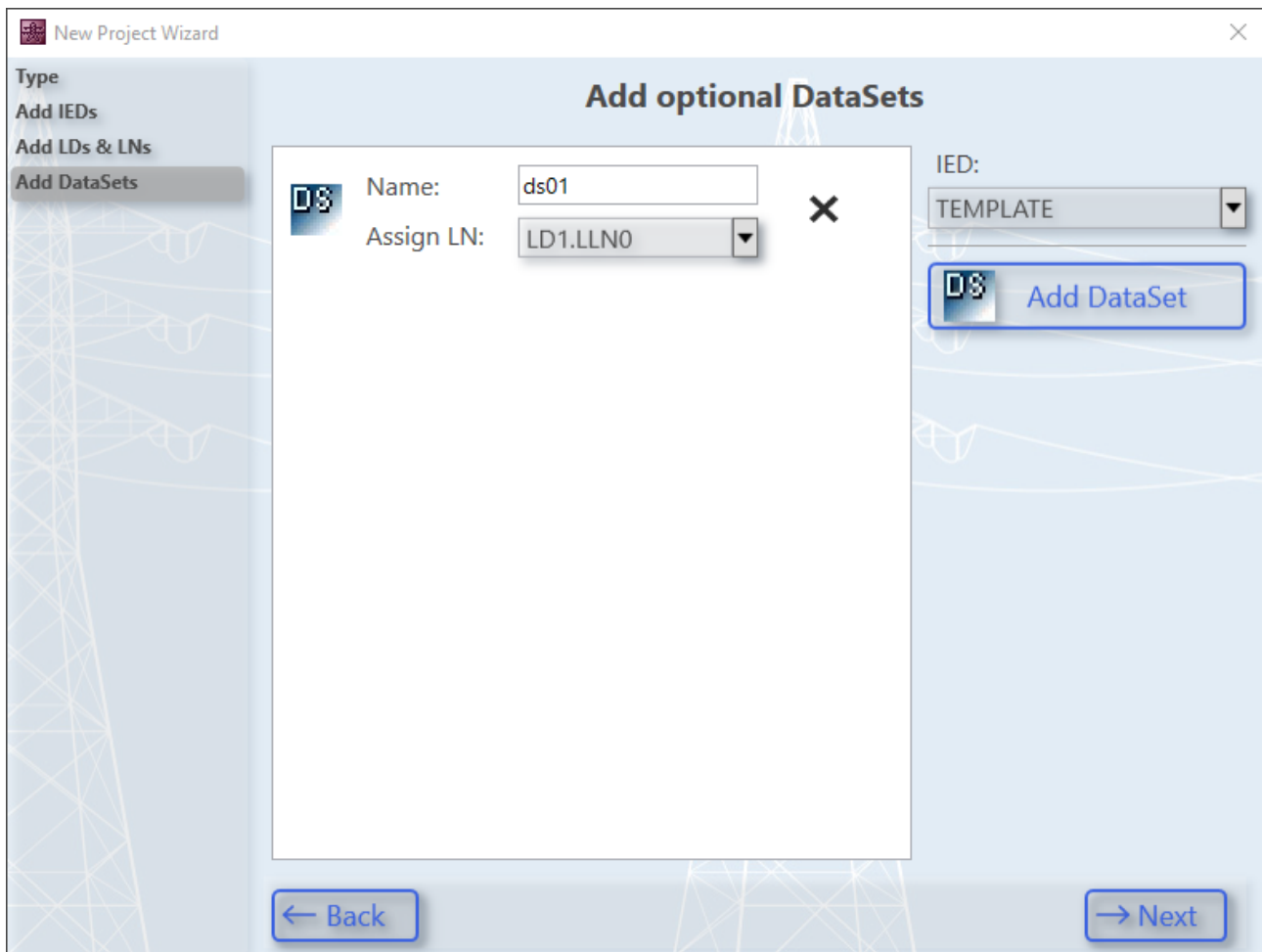
Zudem können Sie beliebige Logical Nodes zum jeweils ausgewählten Logical Device hinzufügen. Bei der Auswahl der Logical Nodes können Sie den Standard auswählen. Nachfolgend die Gruppe der Logical Nodes und den Typ des Knoten. Ein Präfix kann optional vergeben werden.

Jedes Intelligent Electronic Device ist standardmäßig mit einem Logical Device und den beiden erforderlichen Logical Nodes LLN0 und LPHD vorkonfiguriert.



DataSets hinzufügen

Auf der folgenden Seite können Sie beliebige DataSets zum ausgewählten Intelligent Electronic Device hinzufügen. Sie können einen Namen vergeben und das DataSet einem Logical Node zuweisen. Im New Project-Wizard ist es nicht möglich, Functional Constrains und DataAttributes zu den DataSets hinzuzufügen. Dies können Sie nach Abschluss des Wizards im Arbeitsbereich des Konfigurators.



Report Control Blocks hinzufügen

Auf der folgenden Seite können Sie beliebige Report Control Blocks zum ausgewählten Intelligent Electronic Device hinzufügen. Sie haben die Möglichkeit, den Namen und den zugewiesenen Logical Node auszuwählen. Zusätzlich können Sie hier noch jeweils ein DataSet zuweisen, zwischen **Unbuffered** und **Buffered** Report auswählen und den ausgewählten Report Control Block mit dem ausgewählten Maximalwert zu indexieren.

New Project Wizard

Type

- Add IEDs
- Add LDs & LNs
- Add DataSets
- Add Report**

Add optional report control blocks

IED: TEMPLATE

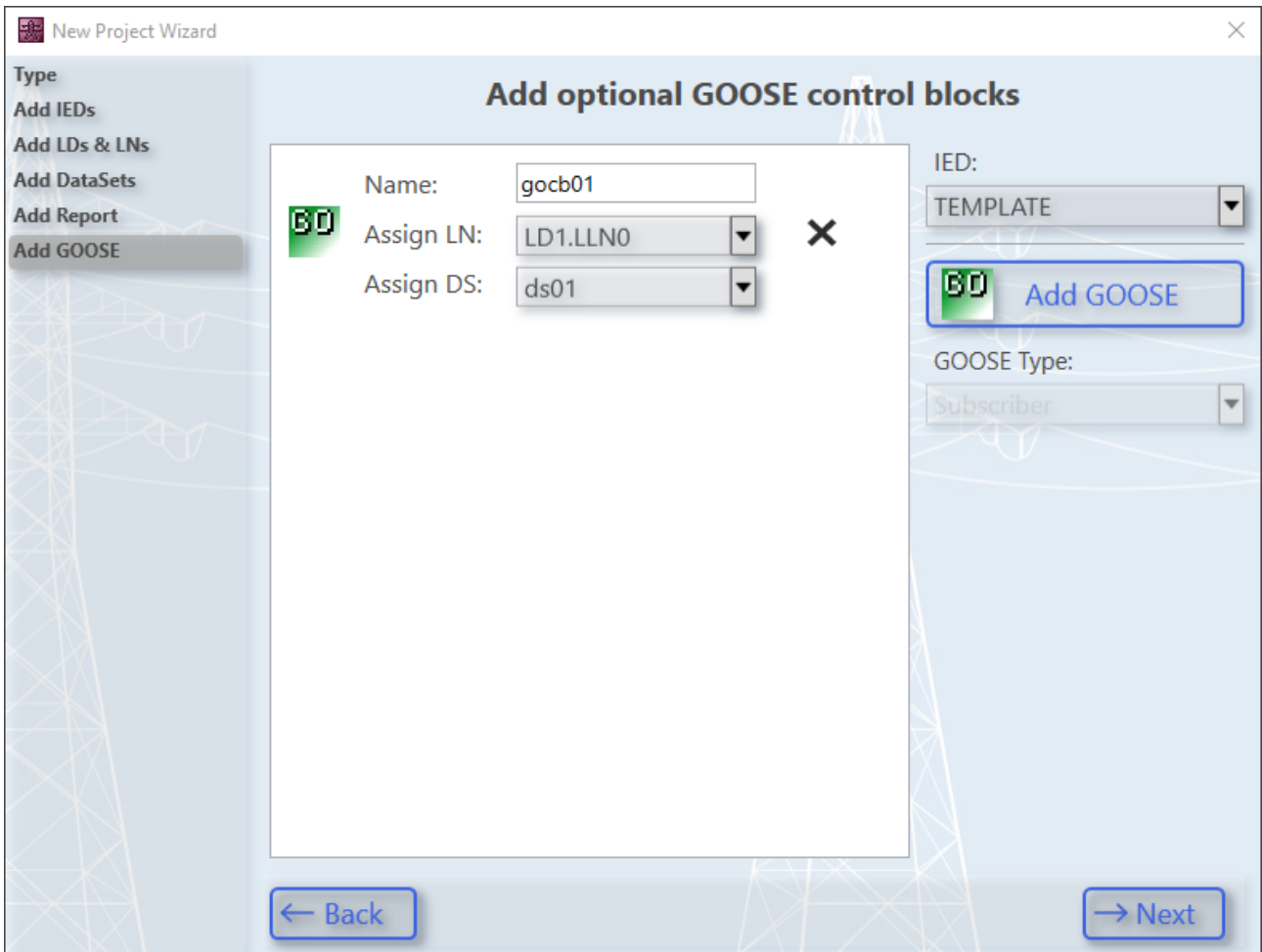
UR Add Report

BR

← Back → Next

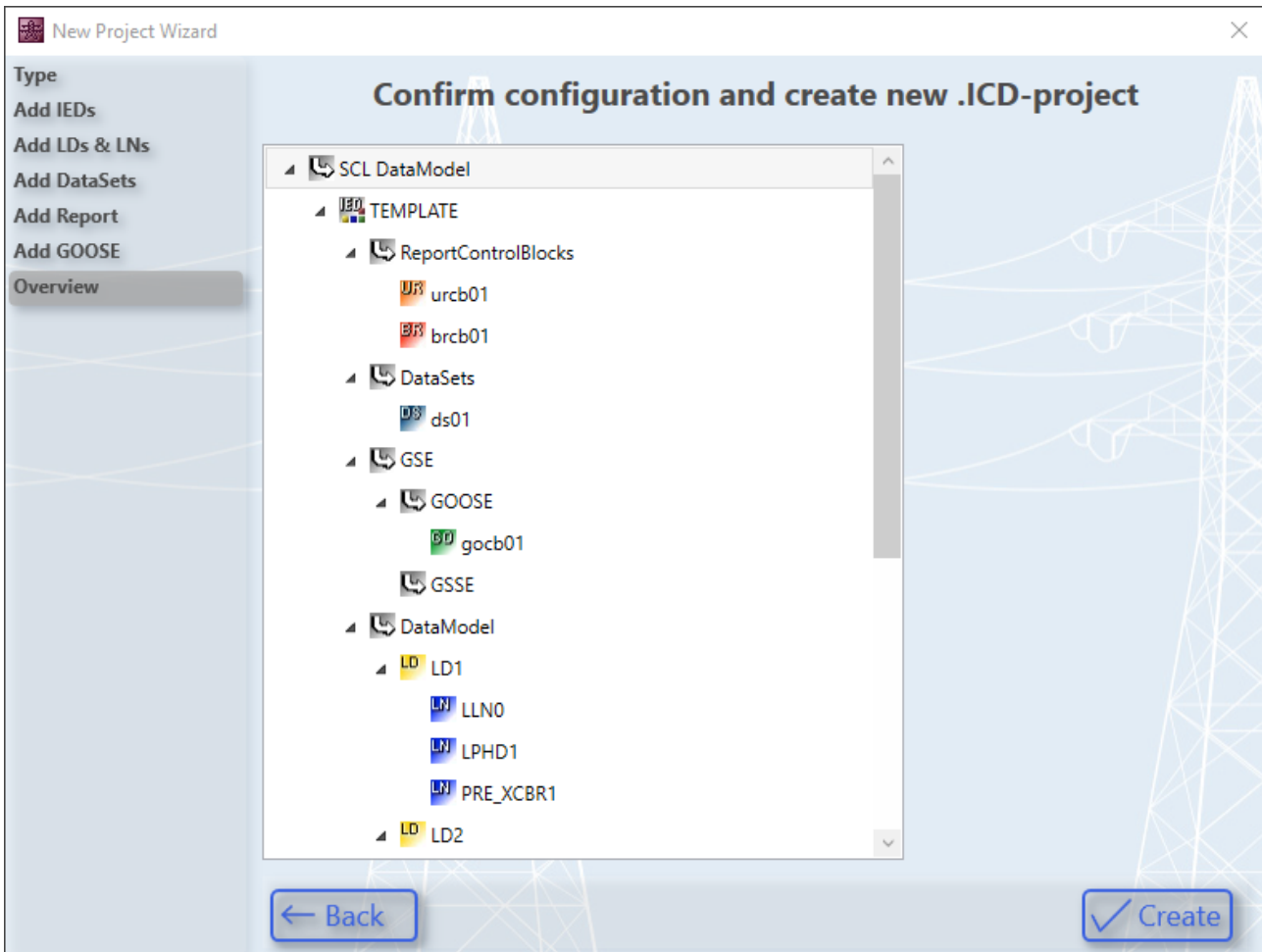
GOOSE Control Blocks hinzufügen

Auf der folgenden Seite können Sie beliebige GOOSE Control Blocks zum ausgewählten Intelligent Electronic Device hinzufügen. Zusätzlich zu den Optionen, den Namen und den zugewiesenen Logical Node auszuwählen, können Sie hier noch jeweils ein DataSet zuweisen. Sobald ein GOOSE Control Block hinzugefügt wurde, wird auf der rechten Seite angezeigt, um welchen GOOSE Typ es sich beim Projekt handelt. Im Falle einer Client-Konfiguration handelt es sich um einen GOOSE Subscriber, bei einer Server-Konfiguration um einen GOOSE Publisher.



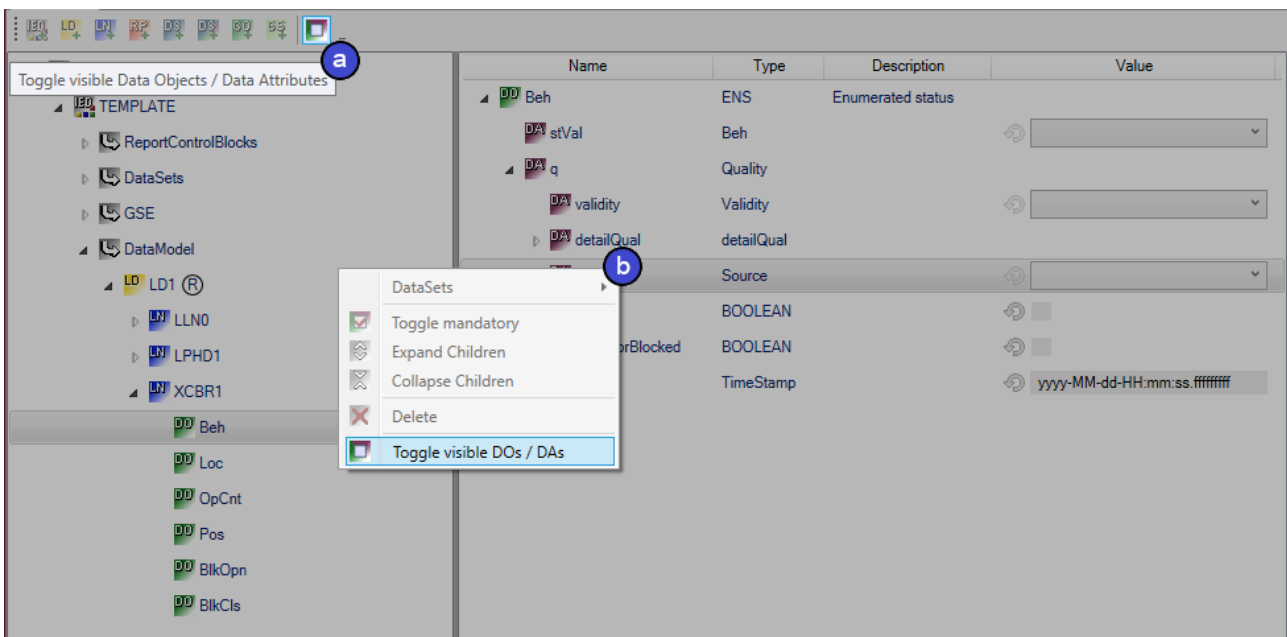
Übersicht

Zum Abschluss des Wizards sehen Sie noch einmal das bisher konfigurierte Datenmodell. Sobald Sie die Vorkonfiguration abgeschlossen haben, können Sie den Create-Button drücken, um die Vorkonfiguration unter dem zuvor ausgewählten Dateinamen zu speichern. Nun können Sie im Konfigurator weitere Einstellungen tätigen oder SPS-Code erzeugen. Alternativ können Sie den Wizard jederzeit über den Close-Button schließen. Die Vorkonfiguration wird in diesem Fall **nicht** gespeichert.

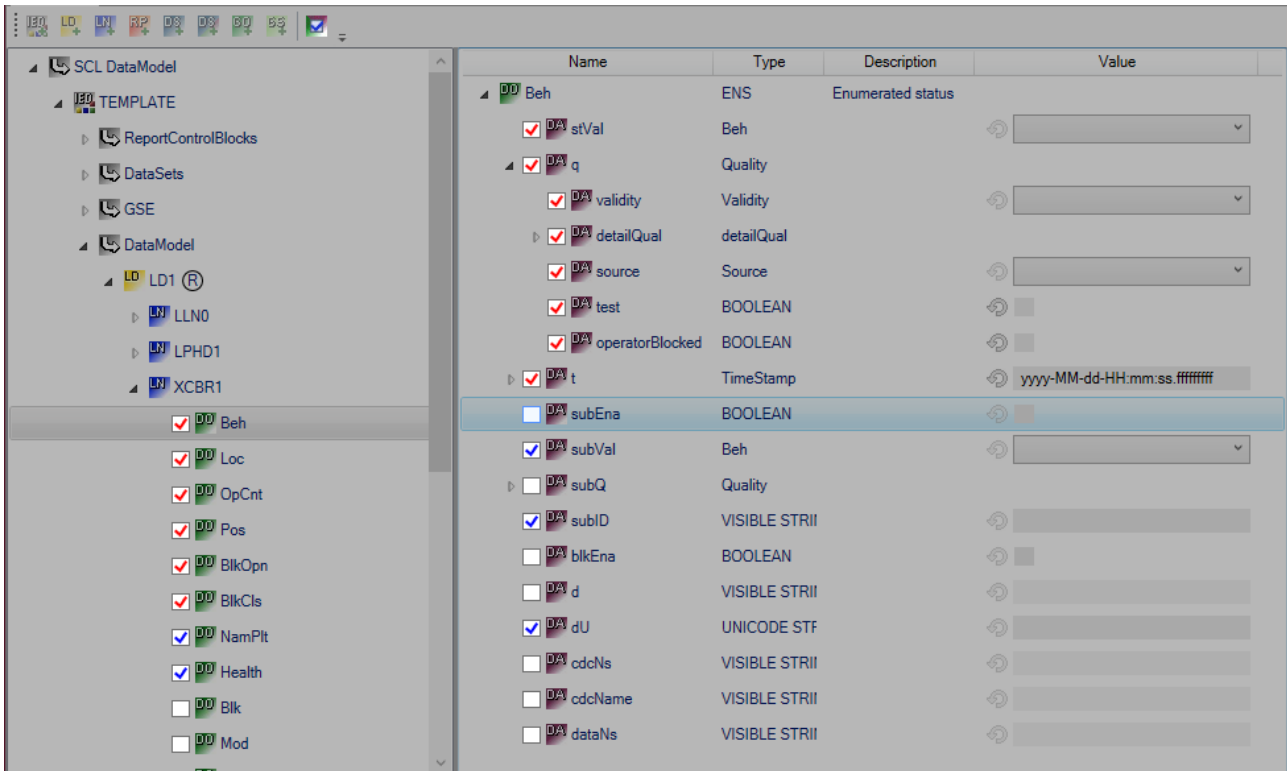


5.4 Konfiguration der Datenobjekte

Im Arbeitsbereich für das Datenmodell können Sie über die Toolbar (a) oder über das Kontextmenü (b) mit dem Kommando **Toggle visible DOs / DAs** alle im Standard zusätzlich für die Knoten definierten Objekte sichtbar machen.



Alle Objekte bekommen darüber hinaus eine Checkbox mit der die optionalen Objekte hinzugefügt oder abgewählt werden können. Der Checkbox-Haken ist bei optionalen Objekten blau, bei erforderlichen Objekten rot. Erforderliche Objekte können über ein Kommando im Kontextmenü des jeweiligen Objektes ebenfalls ein- und ausgeschaltet werden. Davon wird im Sinne des Standards aber abgeraten.

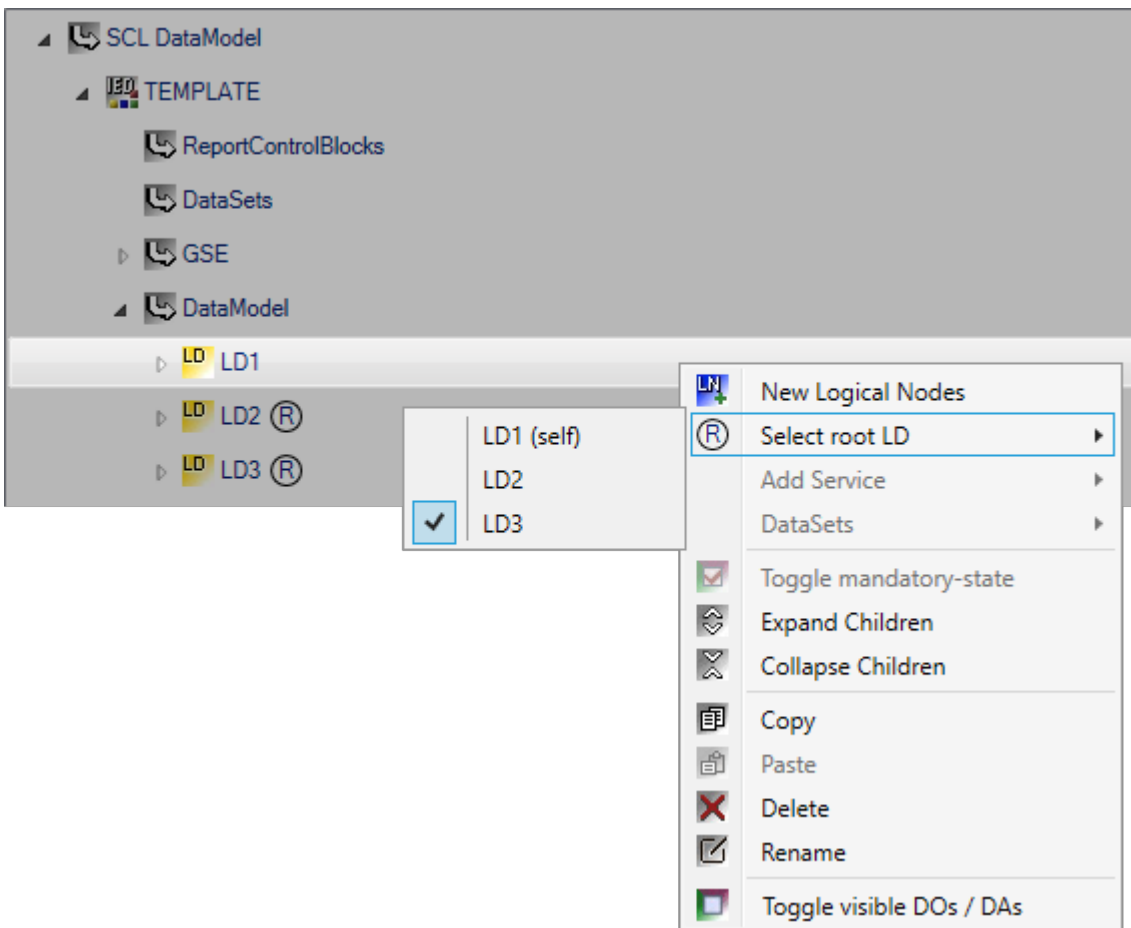


Default-Werte können für die angewählten Attribute in der Spalte Value gesetzt werden. Zur Vermeidung von unbeabsichtigten Eingaben ist dies für abgewählte Attribute nicht möglich. Viele Objekte bieten Enums an, die eine Auswahl per Drop-down-Box ermöglichen. Zahlen und Texte können direkt eingegeben werden. Diese Werte werden später in den SPS-Code exportiert. Der jeweils nebenstehende Zurücksetzen-Button ermöglicht es, den konfigurierten Wert zu löschen.

Name	Type	Description	Value
DD Beh	ENS	Enumerated status	
<input checked="" type="checkbox"/> DA stVal	Beh		blocked
<input checked="" type="checkbox"/> DA q	Quality		
<input checked="" type="checkbox"/> DA validity	Validity		good
<input checked="" type="checkbox"/> DA detailQual	detailQual		
<input checked="" type="checkbox"/> DA source	Source		substituted
<input checked="" type="checkbox"/> DA test	BOOLEAN		✓
<input checked="" type="checkbox"/> DA operatorBlocked	BOOLEAN		✓
<input checked="" type="checkbox"/> DA t	TimeStamp		2021-08-09-10:11:30.426411986
<input checked="" type="checkbox"/> DA SecondSinceEpoch	INT32U		1628503890
<input checked="" type="checkbox"/> DA FractionOfSecond	INT24U		7154006
<input checked="" type="checkbox"/> DA TimeQuality	TimeQuality		
<input type="checkbox"/> DA subEna	BOOLEAN		
<input checked="" type="checkbox"/> DA subVal	Beh		on
<input type="checkbox"/> DA subQ	Quality		
<input checked="" type="checkbox"/> DA subID	VISIBLE STRING64		
<input type="checkbox"/> DA blkEna	BOOLEAN		
<input type="checkbox"/> DA d	VISIBLE STRING255		
<input checked="" type="checkbox"/> DA dU	UNICODE STRING255		Free text...
<input type="checkbox"/> DA cdcNs	VISIBLE STRING255		

Verwaltungshierarchie der Logical Devices

Um komplexe Funktionen zu modellieren, kann es notwendig sein mehrere logische Geräte zu einer Verwaltungshierarchie zu gruppieren. Diese wird mit dem Attribut **setSrcRef** des Objektes **GrRef** im Logical Node **LLN0** aufgebaut, indem über den Wert des Attributes optional ein Logical Device referenziert wird. Wenn in einem Logical Device kein anderes Logical Device referenziert wird, so wird es selbst als Root Logical Device angesehen. Repräsentativ hierfür wird im TwinCAT Telecontrol Konfigurator der Name des eigenen Logical Devices mit einer (**self**) Kennzeichnung versehen, um dies zu verdeutlichen. Das Root Logical Device kann zusätzlich im Kontextmenü des Logical Devices festgelegt werden.



Bedingungsabhängige Datenobjekte

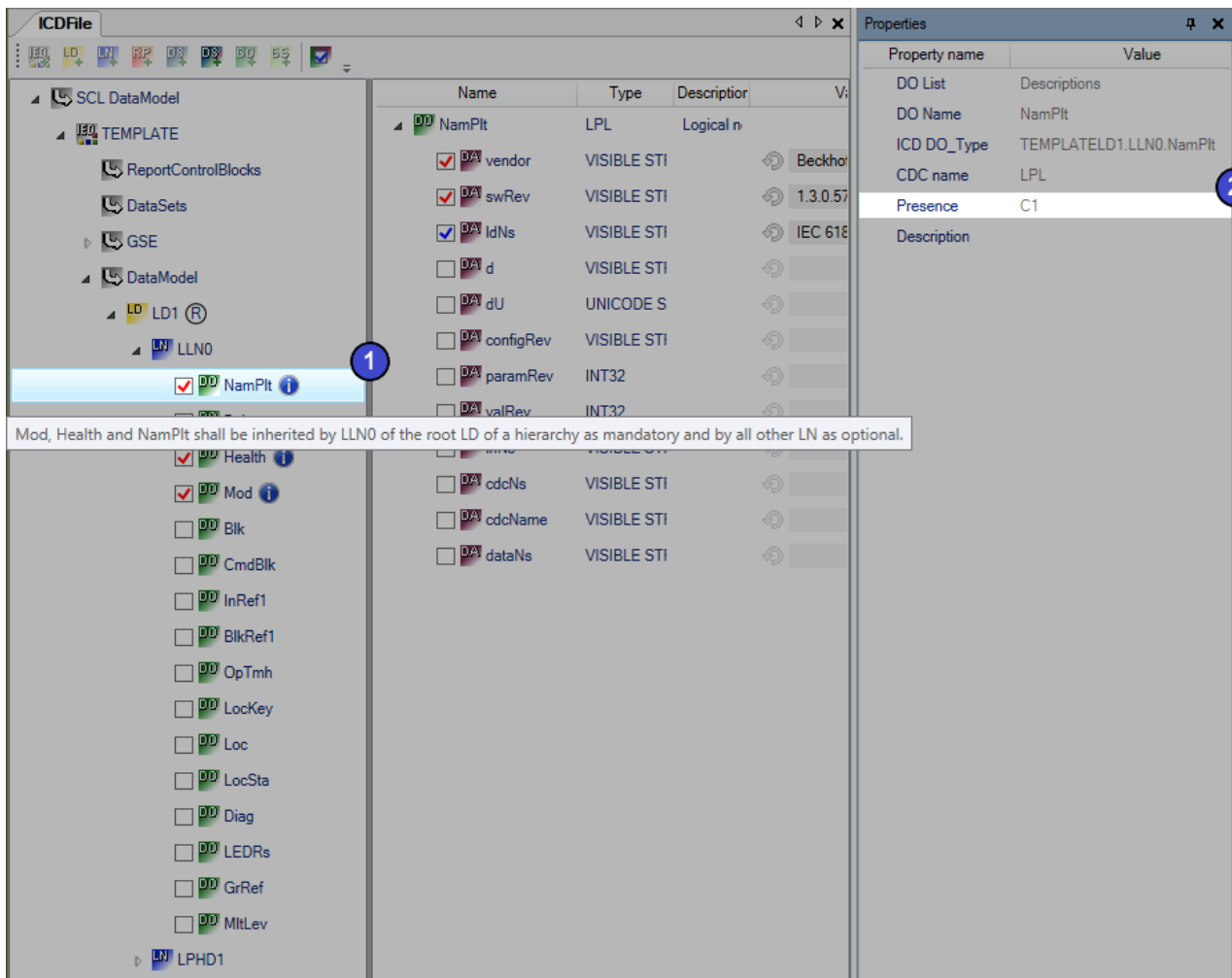
Zusätzlich zu den erforderlichen und optionalen Objekten gibt es noch die **bedingungsabhängigen Datenobjekte**, welche je nach Bedingung im Datenmodell vorhanden, hinzufügbare oder nicht vorhanden sein dürfen. Bedingungsabhängige Datenobjekte werden im Konfigurator je nach Bedingung unterschiedlich behandelt. Falls eine Bedingung nicht automatisch durch den Konfigurator behandelt wird, so wird eine Meldung am jeweiligen Objekt angezeigt. Zusätzlich wird ein ToolTip angezeigt, welcher die jeweilige Bedingungsbeschreibung enthält.

Folgend sehen Sie exemplarisch, wie die Bedingung C1 des Logical Nodes **Common LN** integriert wurde. Dieses Logical Node stellt die Basis aller Logical Nodes dar und wird von diesen geerbt. Im Logical Node **LLN0** werden die drei Objekte **NamPit**, **Health** und **Mod** normkonform als erforderlich hinzugefügt. Im Logical Node **ANCR1** werden sie hingegen nur als optional aufgeführt, da es sich hierbei nicht um das Logical Node **LLN0** des Root Logical Devices handelt.

Mod, Health and NamPit shall be inherited by LLN0 of the root LD of a hierarchy as mandatory and by all other LN as optional.

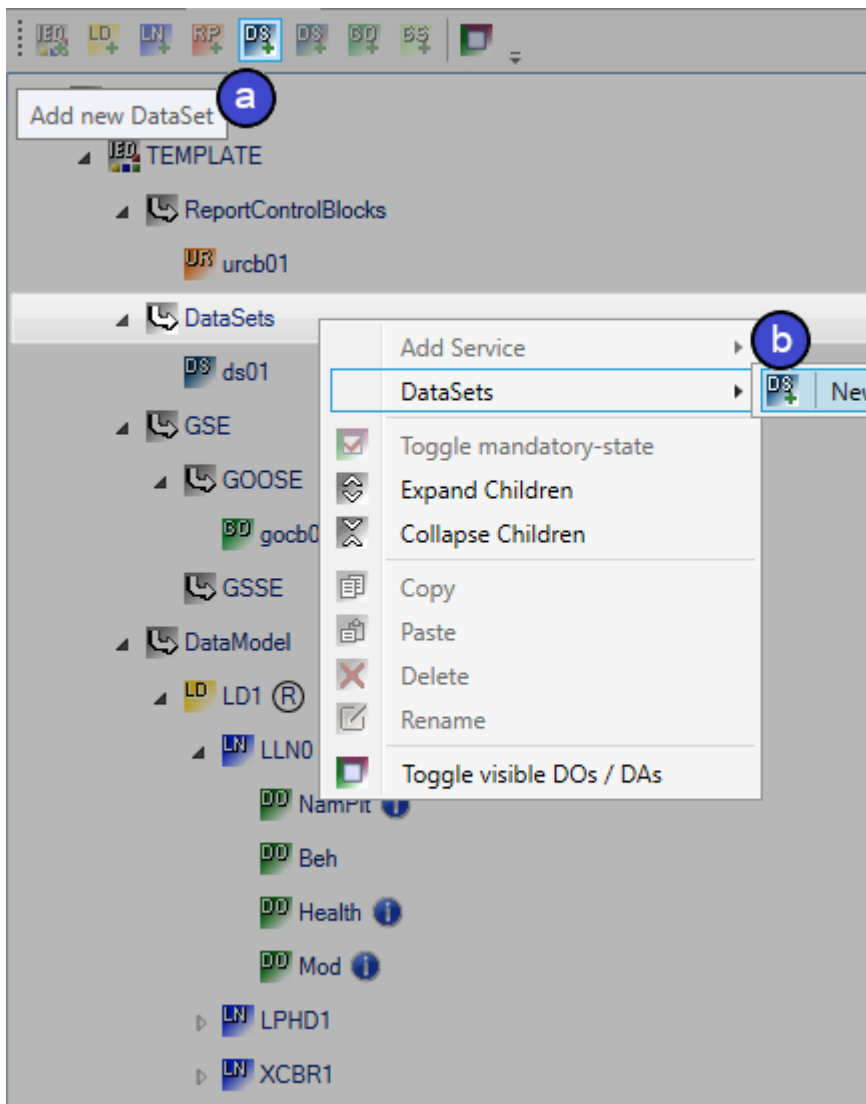
- Health
- Mod
- Blk
- CmdBlk
- InRef1
- BlkRef1
- OpTmh
- LocKey
- Loc
- LocSta
- Diag
- LEDRs
- GrRef
- MitLev
- ▶ LPHD1
- ▶ XCBR1
- ▲ ANCR1
 - NamPit
 - Beh
 - Health
 - Blk
 - Mod

Im Properties-Fenster des jeweiligen Datenobjektes sehen Sie die Bedingungsbezeichnung, unter der Sie die Bedingung im IEC61850-Standard finden können.



5.5 DataSets

DataSets können auf der linken Seite im Datenmodellbaum hinzugefügt werden. Zum einen über die Toolbar (a) und zum anderen über das Kontextmenü (b). Danach öffnet sich ein Dialog, in dem Sie den Namen des DataSets angeben können. Nach Bestätigen des Dialogs wird das **DataSet** in der DataSetlist gelistet.



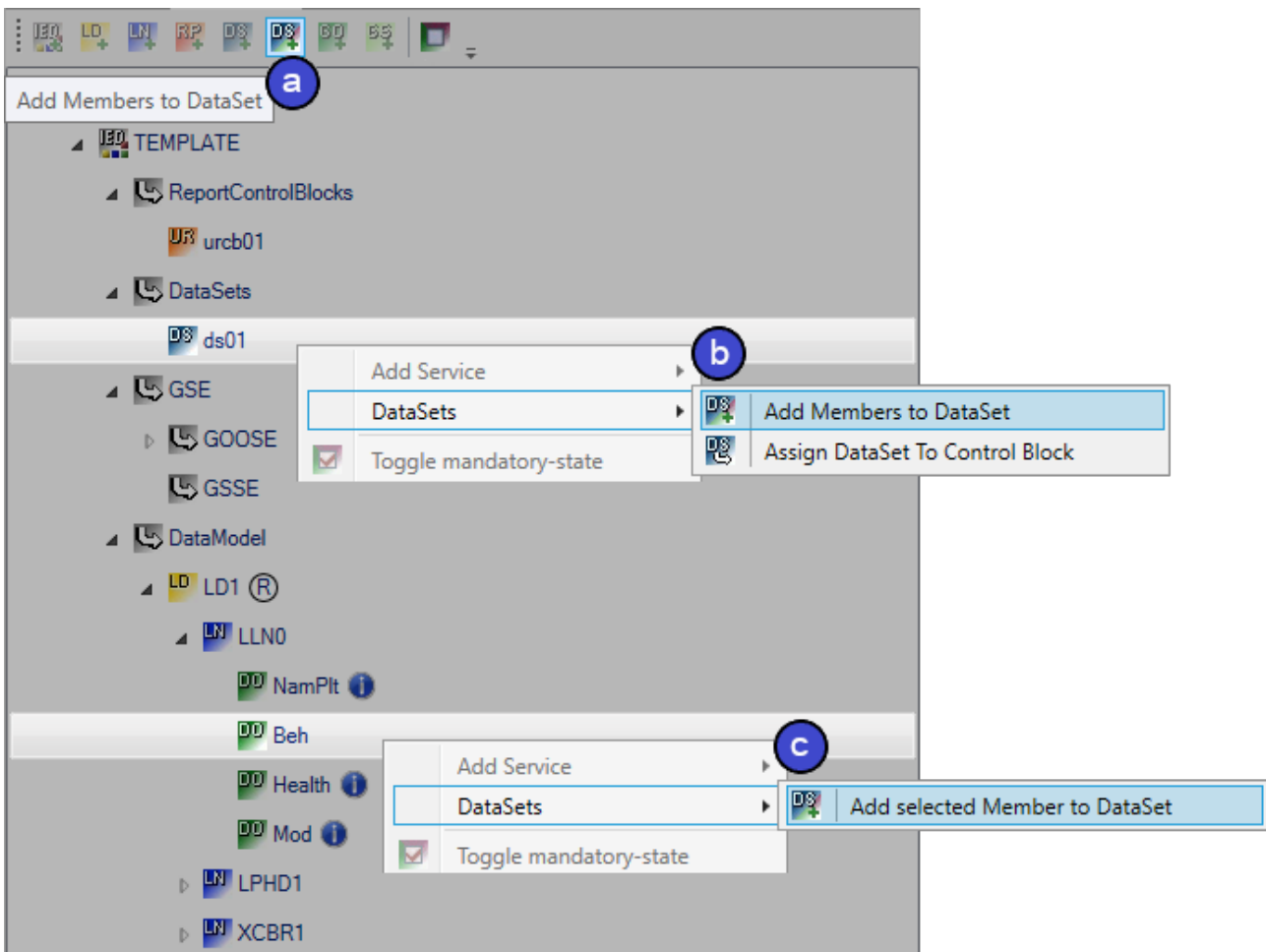
DataSet erstellen

The 'Add DataSet' dialog box is shown with the following fields and buttons:

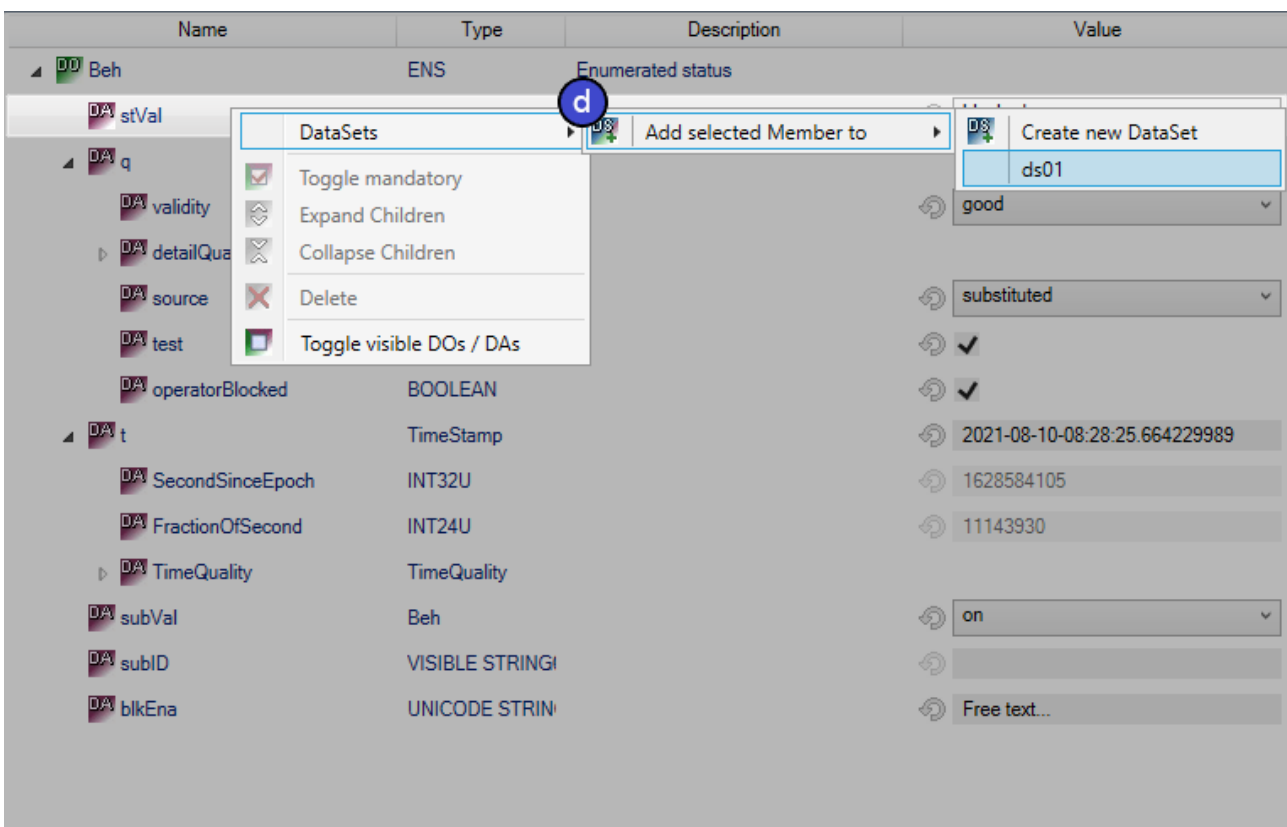
- Logical node:** A dropdown menu currently displaying 'LD1.LLN0'.
- Name:** A text input field containing 'ds02'.
- Buttons:** 'Add' and 'Cancel' buttons at the bottom.

Elemente dem DataSet hinzufügen

DataAttributes und Functional Constrains können dem DataSet über die Toolbar (a) und zum anderen über das Kontextmenü hinzugefügt werden. Je nachdem, welches Element im Datenmodellbaum ausgewählt ist, kann dies über mehrere Möglichkeiten ausgeführt werden.

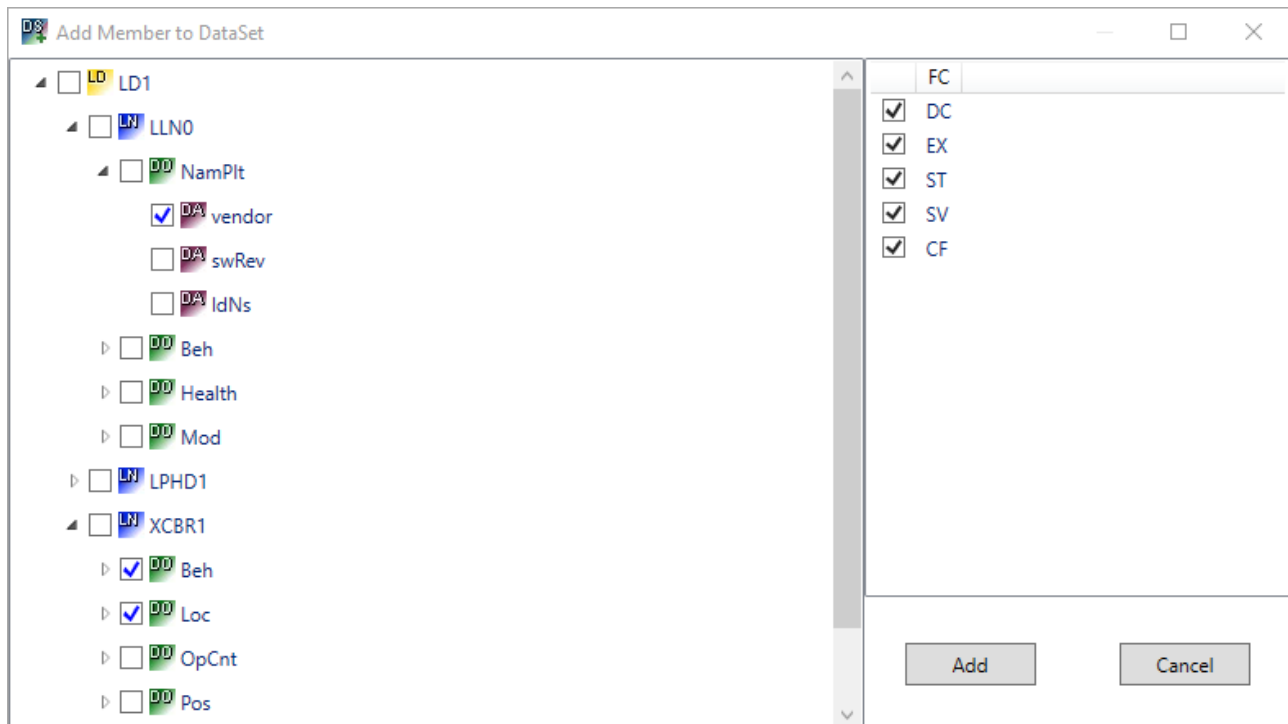


Bei DataAttributes besteht die Möglichkeit, die Zuweisung direkt über das Kontextmenü abzuschließen (d).



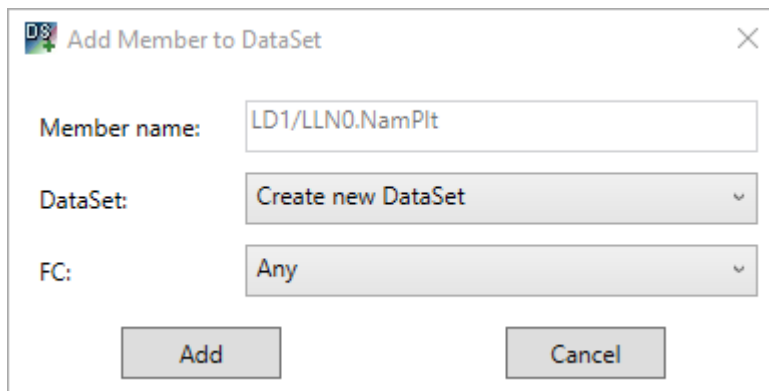
DataSet eigenständig konfigurieren (b)

Mit dem folgenden Dialog ist es möglich, ein DataSet übersichtlich zu konfigurieren, indem Sie die gewünschten Elemente anhaken. Die angezeigten Functional Constrains können Sie über die rechte Spalte filtern.



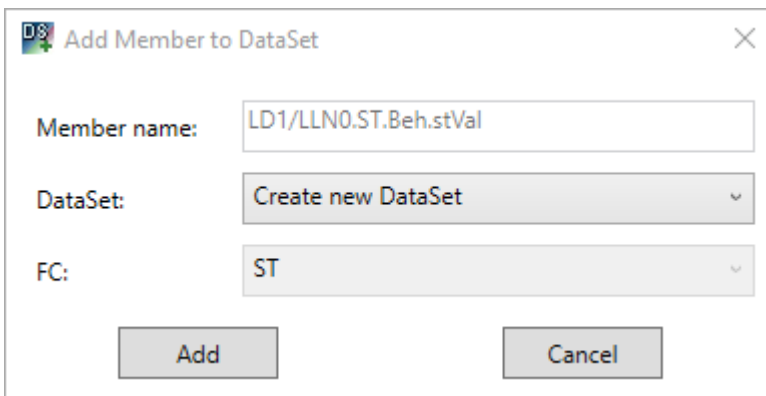
Functional Constrains hinzufügen (c)

Im folgenden Dialog können Sie auswählen, zu welchem neuen oder bestehenden DataSet der Functional Constrain hinzugefügt werden soll. Sie können zudem entscheiden, einen bestimmten Functional Constrain oder alle im DataObject vorhandenen Functional Constrains hinzuzufügen.

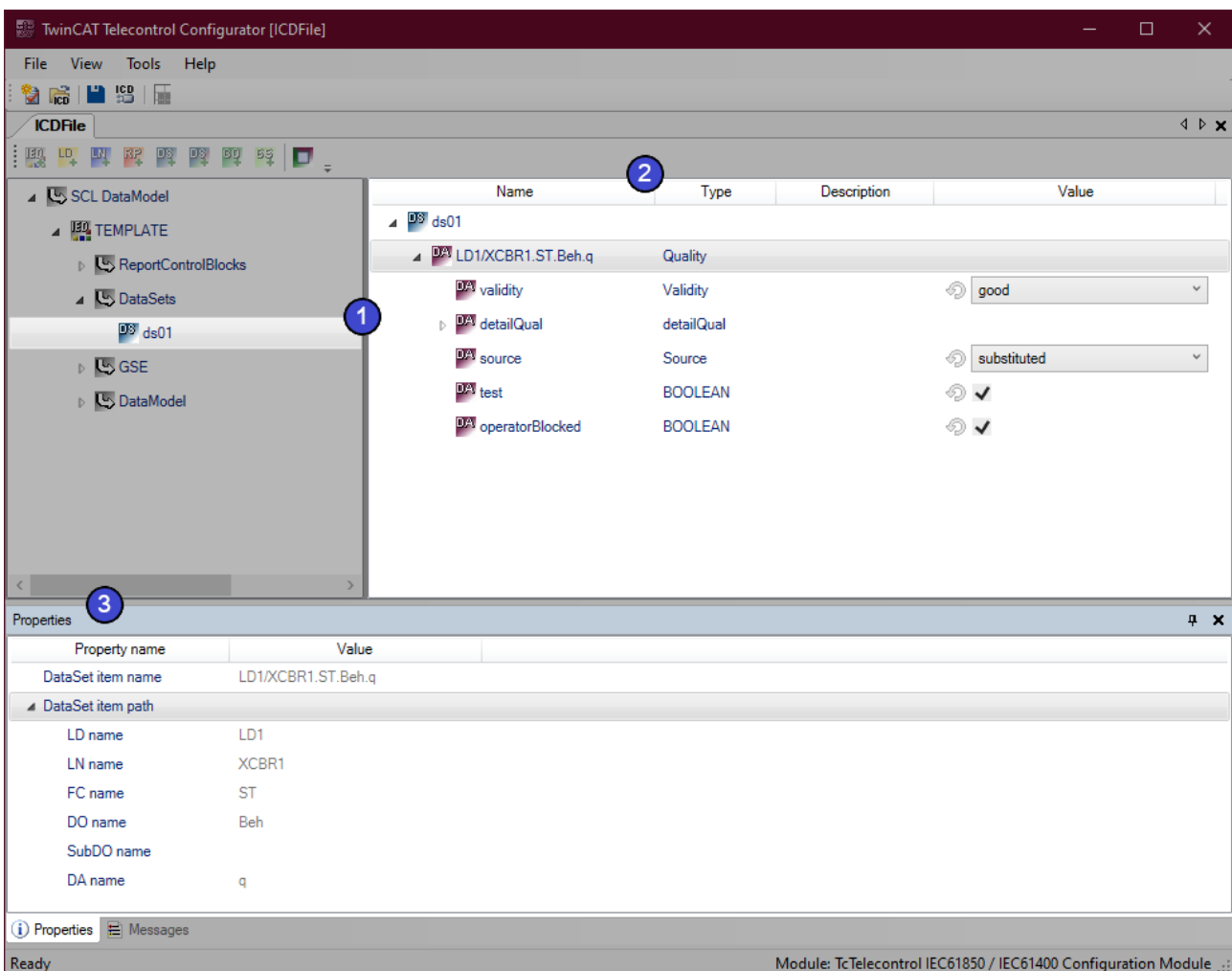


DataAttributes hinzufügen (d)

Sie können zusätzlich zur Konfiguration über das Kontextmenü auch ein DataAttribute über die Toolbar zum DataSet hinzufügen. Hierbei wird der folgende Dialog geöffnet, in dem Sie auswählen können, zu welchem neuen oder bestehenden DataSet das DataAttribute hinzugefügt werden soll.



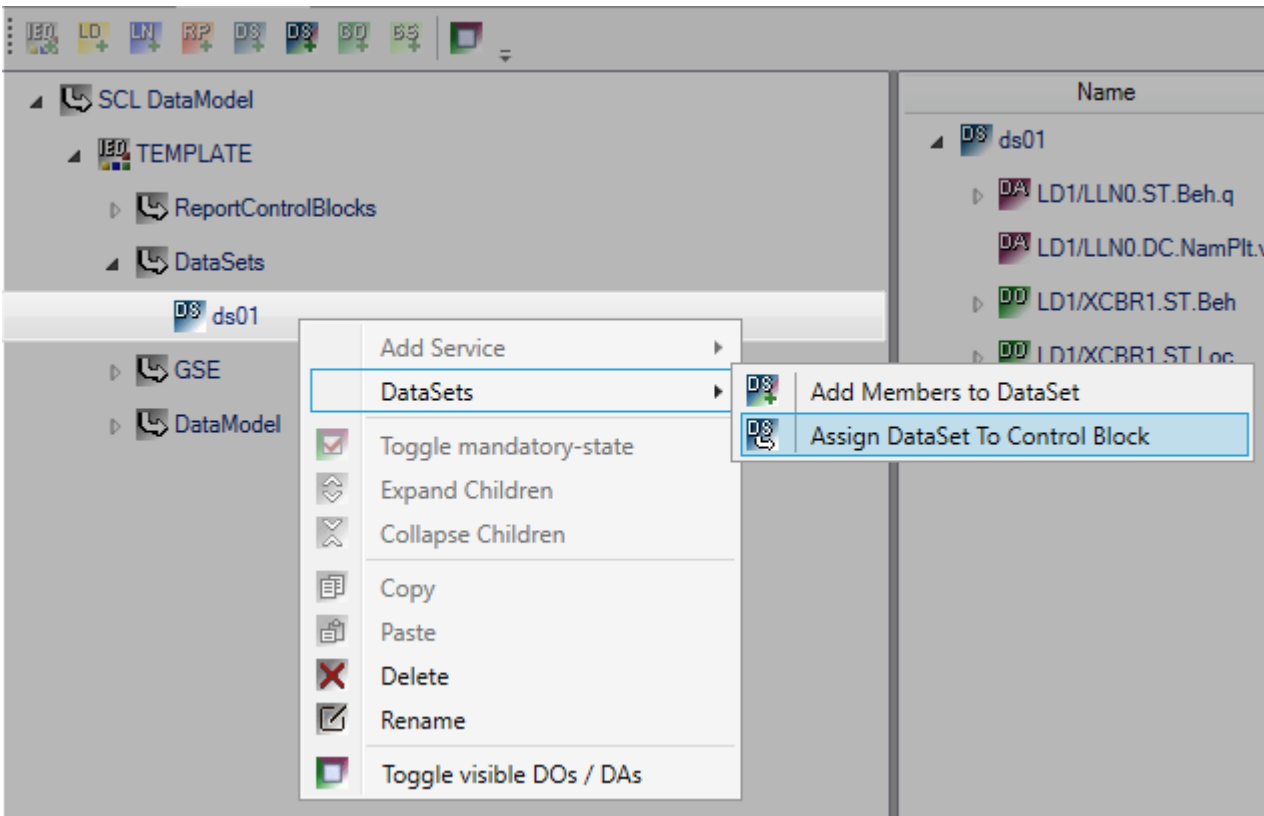
Wählen Sie ein **DataSet** aus der DataSetlist aus (1), so werden alle dem **DataSet** zugehörigen DataAttributes und Functional Constrains im rechten Fenster dargestellt (2).



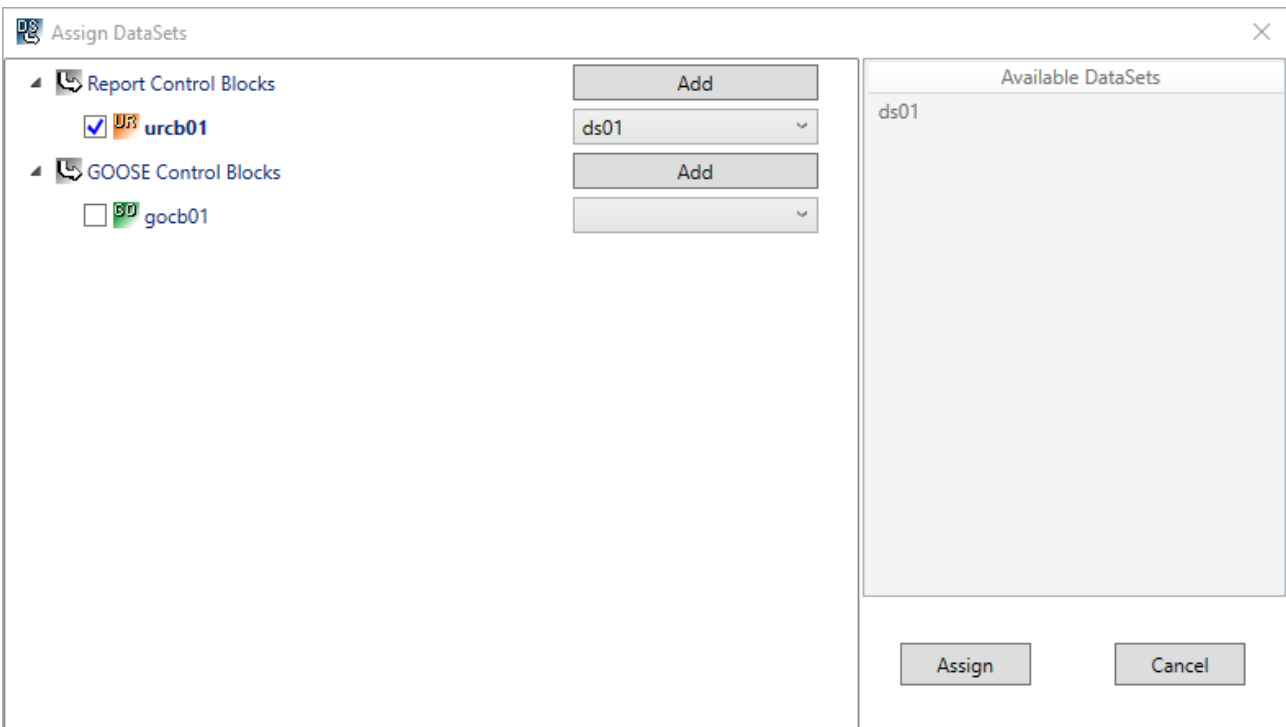
In den Properties ist die Quelle des gewählten Attributes dokumentiert (3).

DataSets Kontrollblöcken zuweisen

DataSets können zum einen direkt in den Eigenschaften des jeweiligen Kontrollblockes verknüpft werden. Zum anderen können Sie das jeweilige DataSet direkt über das zugehörige Kontextmenü mehreren Kontrollblöcken zuweisen.

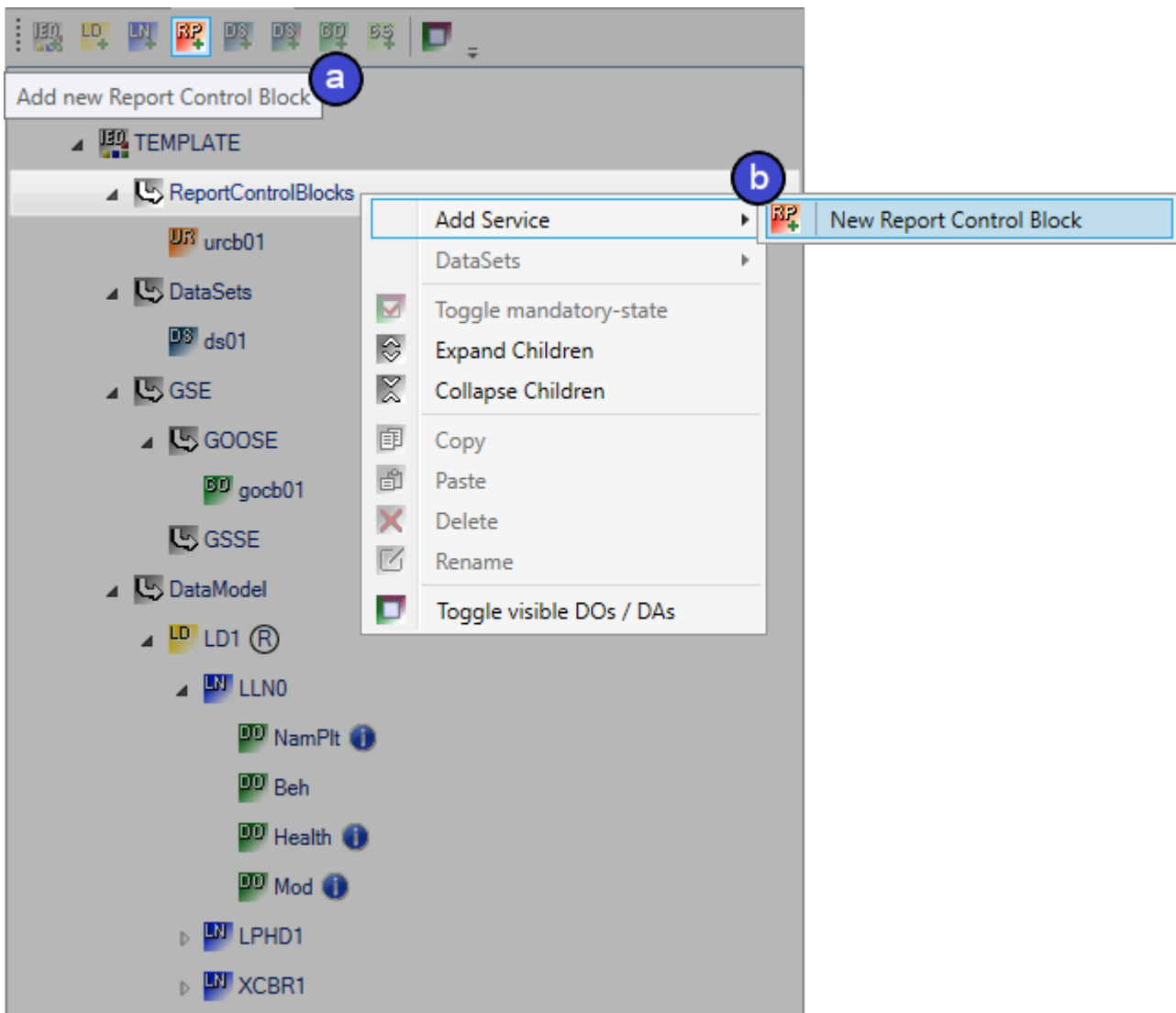


Der folgende Dialog zeigt alle vorhandenen Kontrollblöcke und das jeweils aktuell zugewiesene DataSet. Zudem besteht die Möglichkeit, neue Kontrollblöcke hinzuzufügen. Um das im Datenmodellbaum ausgewählte DataSet nun einem Kontrollblock zuzuweisen, können Sie diesen direkt anhaken. Alternativ ist es möglich, über das Auswahlfeld andere, bereits vorhandene DataSets den jeweiligen Kontrollblöcken zuzuweisen. In der rechten Spalte werden alle aktuell vorhandenen DataSets angezeigt.

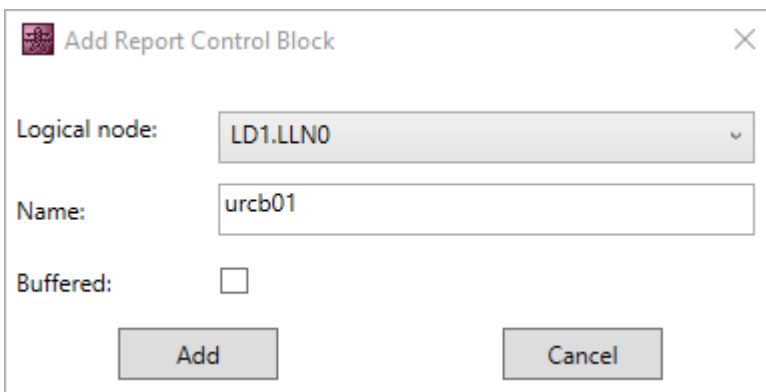


5.6 Reporting

Reportingblöcke (**Report Control Block**) können über einen Button in der Toolbar erzeugt werden (a), alternativ aus dem Kontextmenü heraus im Datenmodellbaum unter ReportControlBlocks (b).



Es besteht die Möglichkeit **Buffered** oder **Unbuffered** Reportblöcke zu erzeugen. Die Zuordnung zu einem Logical Node kann in dem Dialog ebenso gewählt werden, wie ein entsprechender Name.



Wenn Sie einen angelegten Report Control Block im Baum anwählen, können Sie Einstellungen in den Properties vornehmen. Je nachdem ob es sich um einen unbuffered oder buffered Report Control Block handelt, stehen unterschiedliche Standard-Einstellungen zur Verfügung.

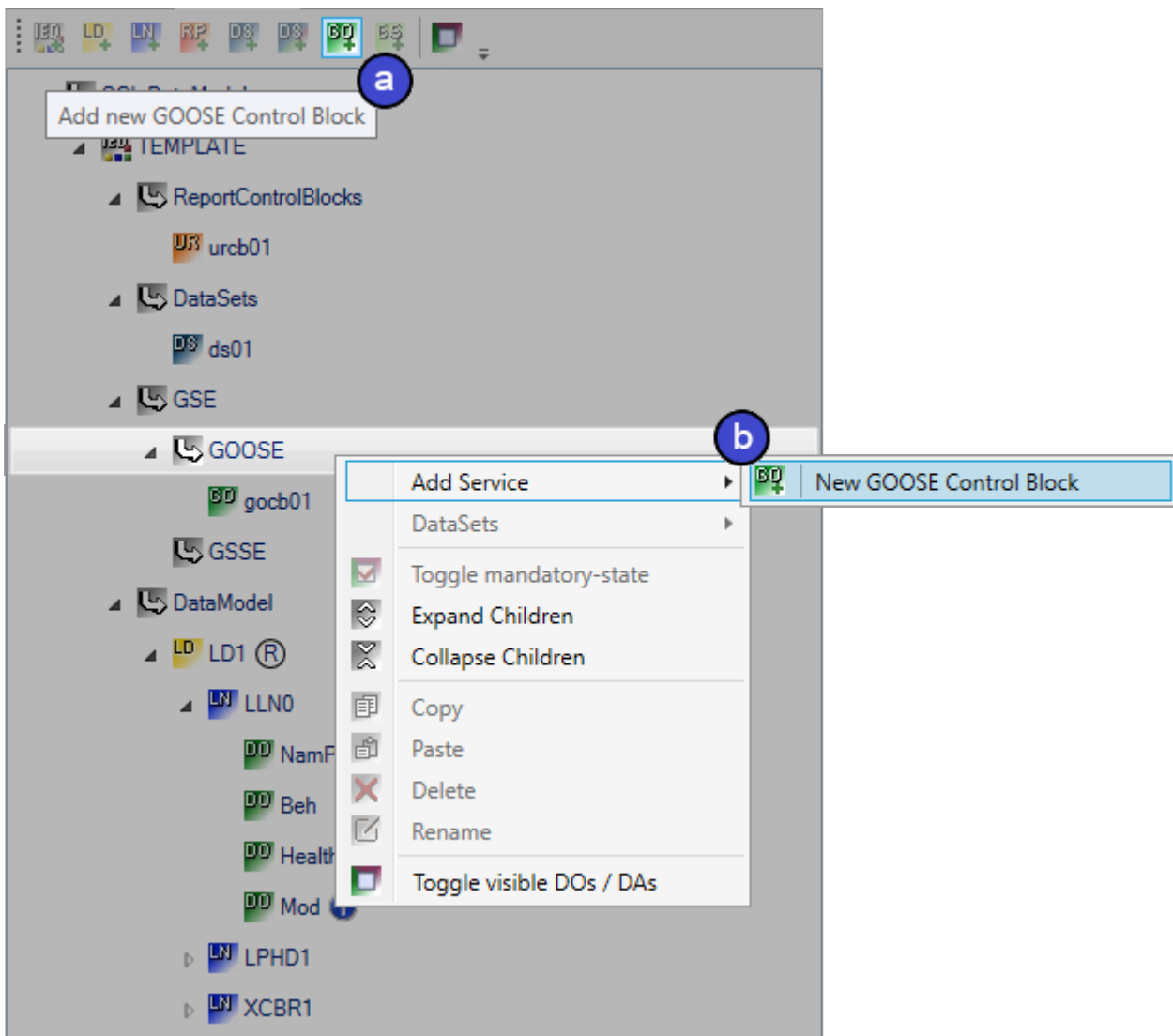
Zusätzlich zu den Standard-Einstellungen gibt es noch Einstellungen für den SPS Server. Weitere Informationen hierzu finden Sie im Kapitel [Server - Buffered Reporting](#) [▶ 662].

Properties	
Property name	Value
Name	urcb01
Buffered	False
Resv	False
DataSet Name	ds01
Integrity Period	0
▲ Trigger Options	
dchg	False
qchg	False
dupd	False
period	False
gi	True
▲ Optional Fields	
seqNum	False
timeStamp	False
reasonCode	False
dataSet	False
dataRef	False
bufOvfl	False
entryID	False
configRef	False
segmentation	False
Report ID	
Report Enabled	False
Configuration Revision	0
Buffer Time	0
Indexed	False
Sequence Number	0
GI	False
Owner	
Referenced LN	TEMPLATELD1/LLN0
▲ Server PLC settings	
Overwrite changes in same cyc	False
Overwrite buffered changes for	False
Description	

Properties	
Property name	Value
Name	brcb01
Buffered	True
DataSet Name	ds01
Integrity Period	0
▲ Trigger Options	
dchg	False
qchg	False
dupd	False
period	False
gi	True
▲ Optional Fields	
seqNum	False
timeStamp	False
reasonCode	False
dataSet	False
dataRef	False
bufOvfl	True
entryID	False
configRef	False
segmentation	False
Report ID	
Report Enabled	False
Configuration Revision	0
Buffer Time	0
Reserve Time	0
Indexed	False
Sequence Number	0
GI	False
Purge Buffer	False
Entry ID	0
Time Of Entry	
Owner	
Referenced LN	TEMPLATELD1/LLN0
▲ Server PLC settings	
Max buffer entries	10
Overwrite changes in same cyc	False
Overwrite buffered changes for	False
Description	

5.7 GOOSE

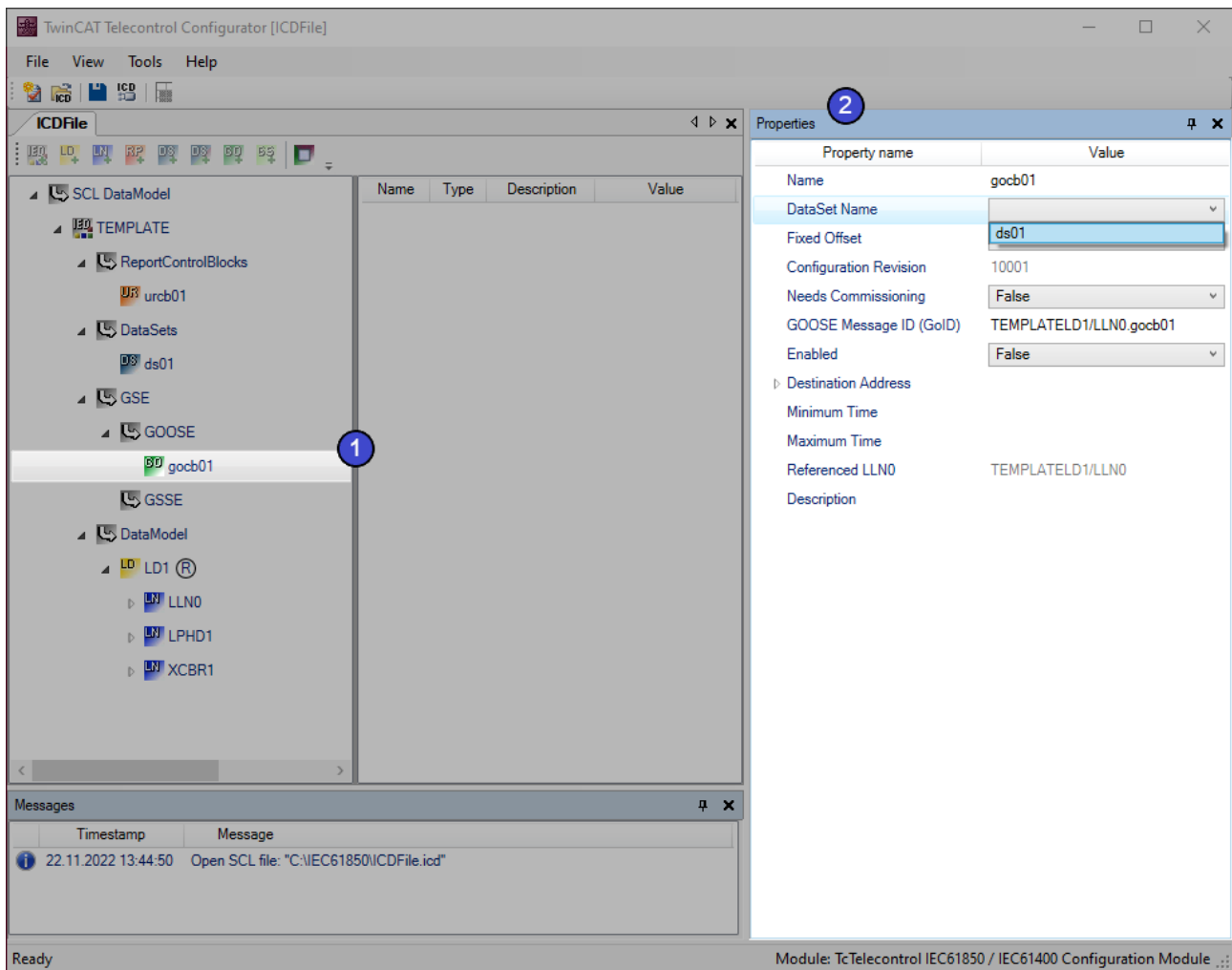
GOOSE Blöcke (**GOOSE Control Block**) können über einen Button in der Toolbar erzeugt werden (**a**), alternativ aus dem Kontextmenü heraus im Datenmodellbaum unter GOOSE (**b**).



Die Zuordnung zu einem Logical Node können Sie in dem Dialog ebenso wählen, wie ein entsprechender Name.



Wenn Sie den im Baum angelegten GOOSE Block selektieren (1), können Sie GOOSE-spezifische Einstellungen in den Properties vornehmen und so zum Beispiel die Verknüpfung zu einem der angelegten DataSets realisieren (2).

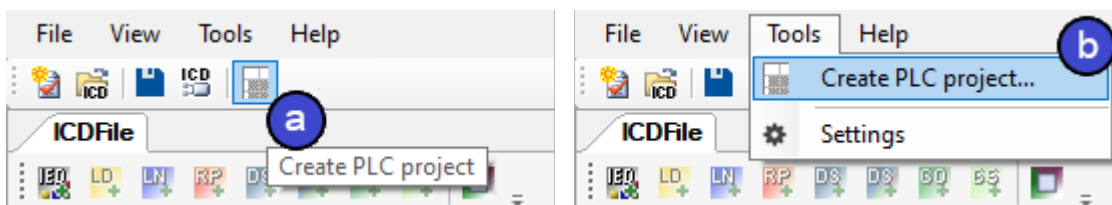


5.8 Automatische Code-Erzeugung

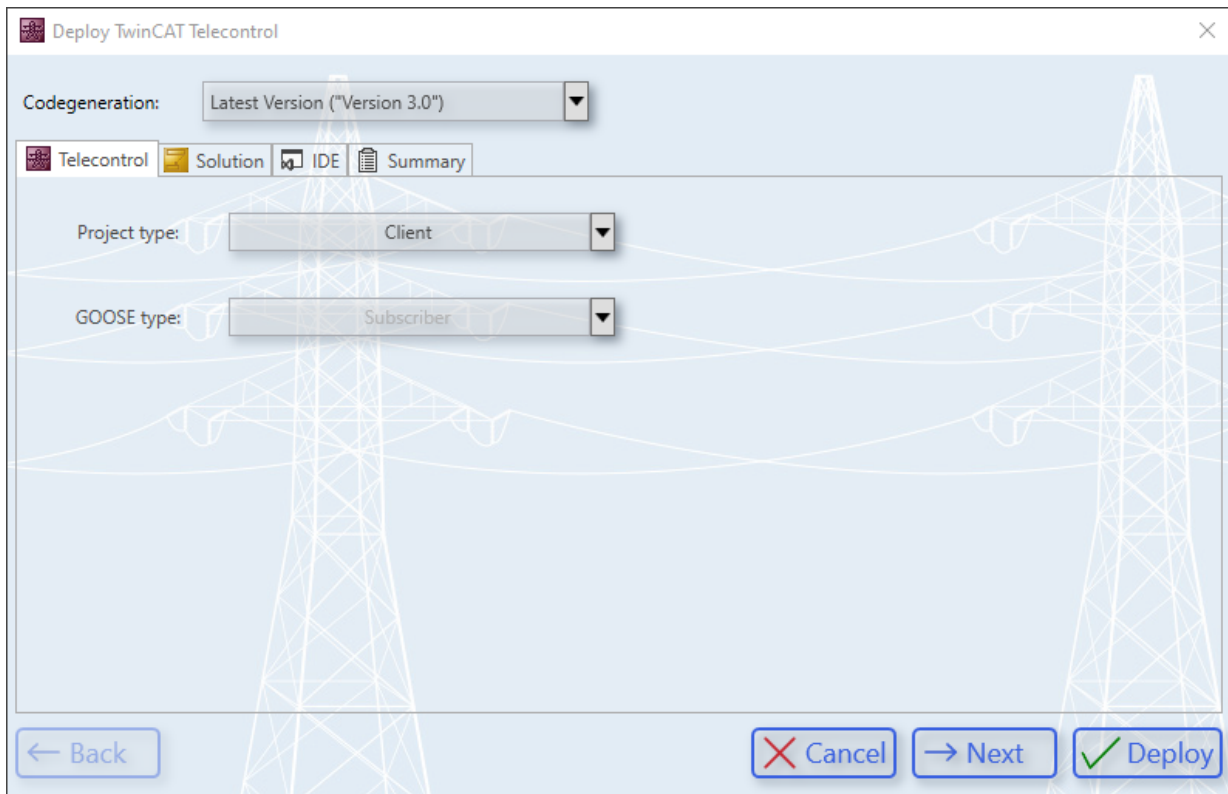
● Automatische SPS Code Generierung

i Der von dem TwinCAT Telecontrol Configurator erzeugte Code ist lediglich ein Anwendungsbeispiel, welches auch von Version zu Version angepasst werden kann. Um mit den Funktionsbausteinen der IEC 61850 eine funktionsfähige Applikation zu realisieren, sind mehrere Wege möglich.

- ✓ Die automatische SPS-Code-Erzeugung aus dem TwinCAT Telecontrol Configurator erfolgt über das TwinCAT Automation Interface.
- 1. Durch die Betätigung des Buttons **Create PLC Project** in der Toolbar(a) oder im Tools-Menü des Konfigurators(b) öffnet sich der Deploy TwinCAT Telecontrol-Dialog.

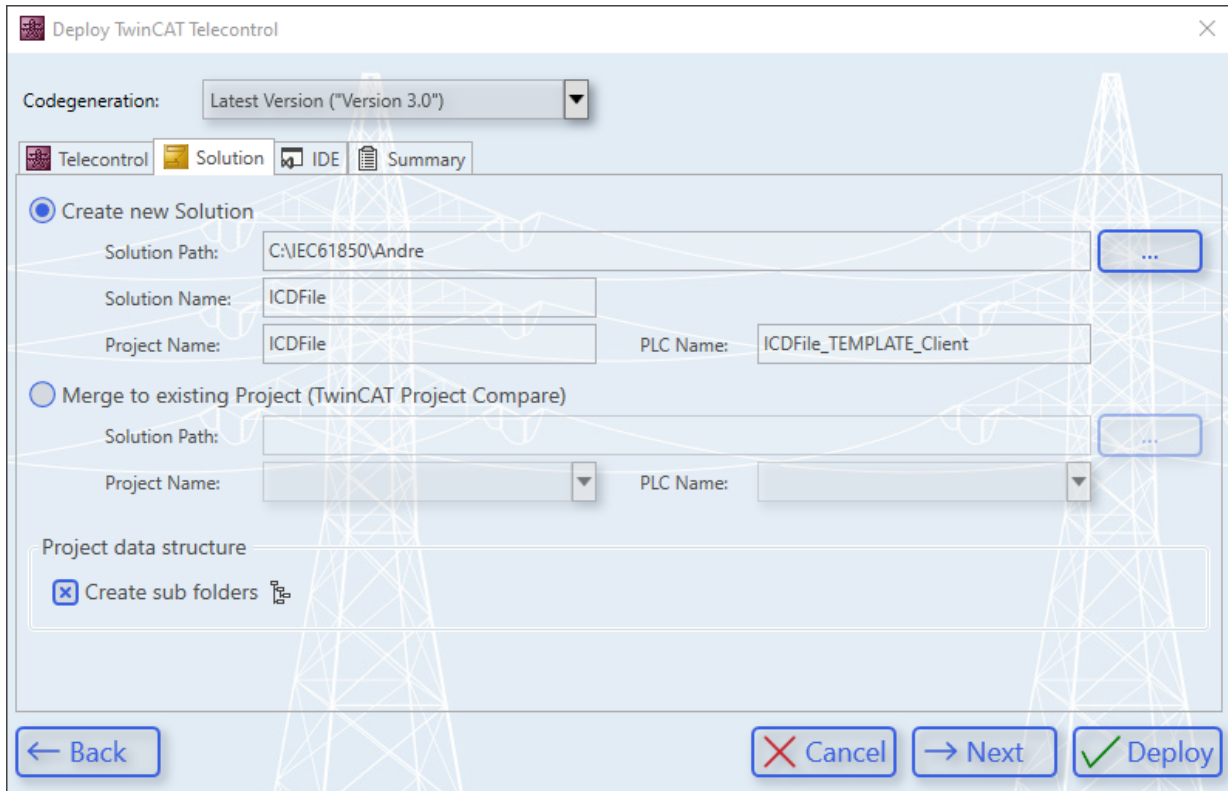


2. Im Tab **Telecontrol** haben Sie die Möglichkeit, zwischen Client- und Serverkonfiguration zu wählen (**Project Type**). Falls GOOSE Control Blocks im zu generierenden Projekt vorhanden sind, so wird auch automatisch der zugehörige **GOOSE Typ** ausgewählt.

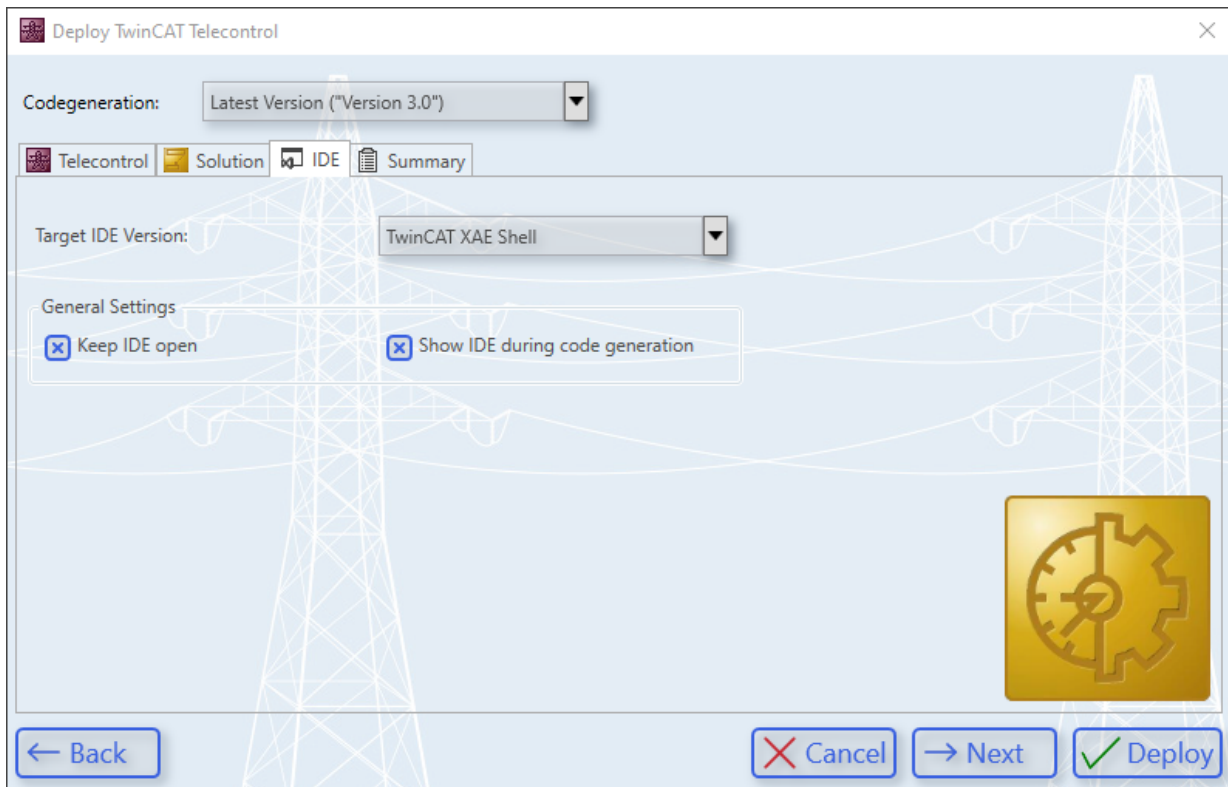


3. Im Tab **Solution** wählen Sie einen Pfad aus (**Solution Path**) und vergeben die Namen für **Solution**, **XAE-Projekt** und **SPS-Projekt**. Hierbei können Sie entweder eine neue Solution erstellen oder die Änderungen mit Hilfe des TwinCAT Project Compare-Tools in eine bestehende Solution einbinden. Wählen Sie hierzu zunächst die Option **Merge to existing Project** und danach eine bestehende SLN-Datei mit dem nebenstehenden Button „...“aus.
Unter **Project data structure** können Sie zusätzlich auswählen, ob Unterordner (**Create sub folders**) generiert werden sollen. Standardmäßig werden diese generiert, um in der Ordnerhierarchie das konfigurierte Datenmodell abzubilden.

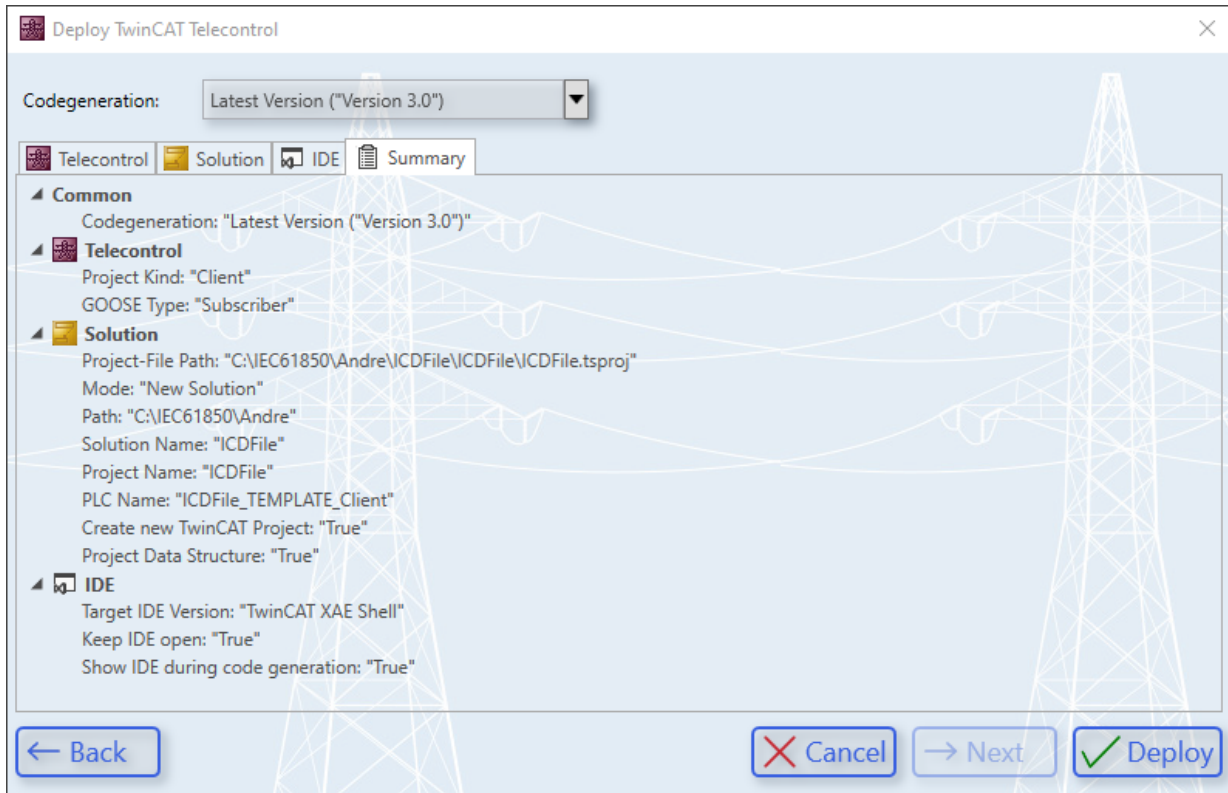
Wenn Sie längere Namen für die Komponenten des Datenmodells verwenden, kann es notwendig sein, keine Unterordner zu generieren, um die maximale Pfadlänge nicht zu überschreiten.



4. Im Tab **IDE** haben Sie die Möglichkeit, eine Entwicklungsumgebung ihrer Wahl zur SPS-Code-Generierung zu nutzen (**Target IDE Version**). Voraussetzung hierfür ist eine vorhandene TwinCAT-Integration in der Entwicklungsumgebung. Zudem können Sie einstellen, ob die Entwicklungsumgebung nach abgeschlossener SPS-Code-Generierung geöffnet bleiben soll (**Keep IDE open**) und ob die Entwicklungsumgebung während der SPS-Code-Generierung sichtbar sein soll (**Show IDE during code generation**).

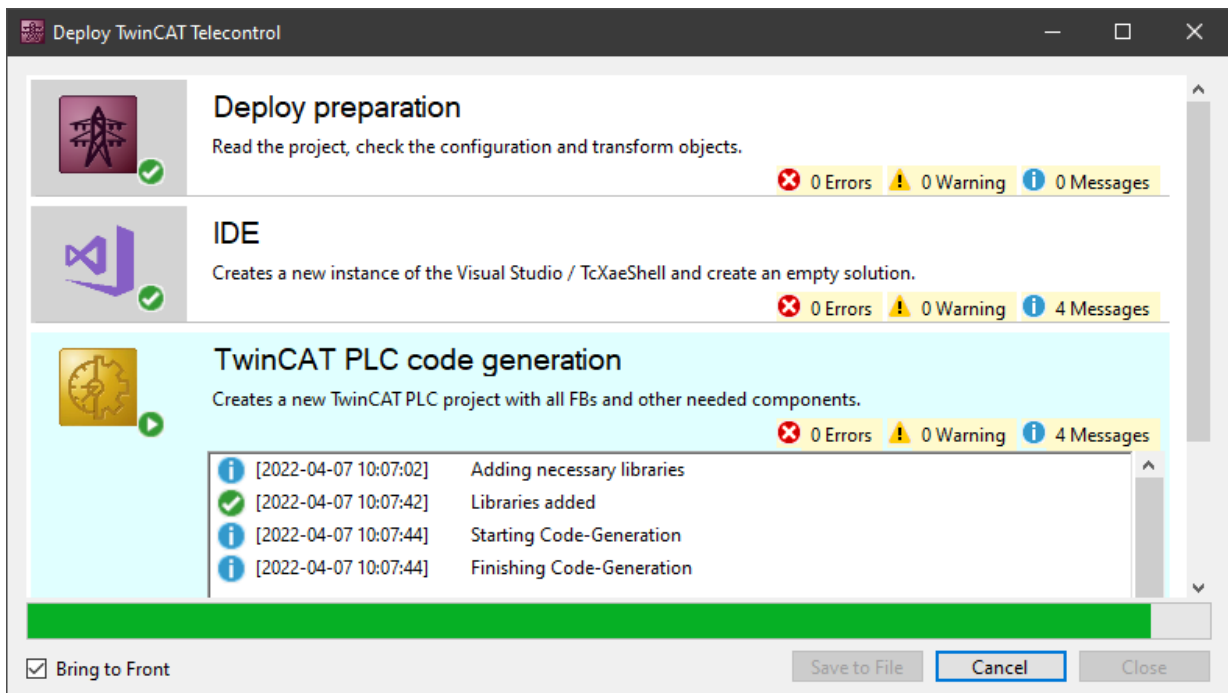


- Das letzte, abschließende Tab **Summary** bietet Kurzübersicht der getätigten Einstellungen. Die SPS-Code-Generierung wird gestartet, sobald Sie den Button **Deploy** drücken.

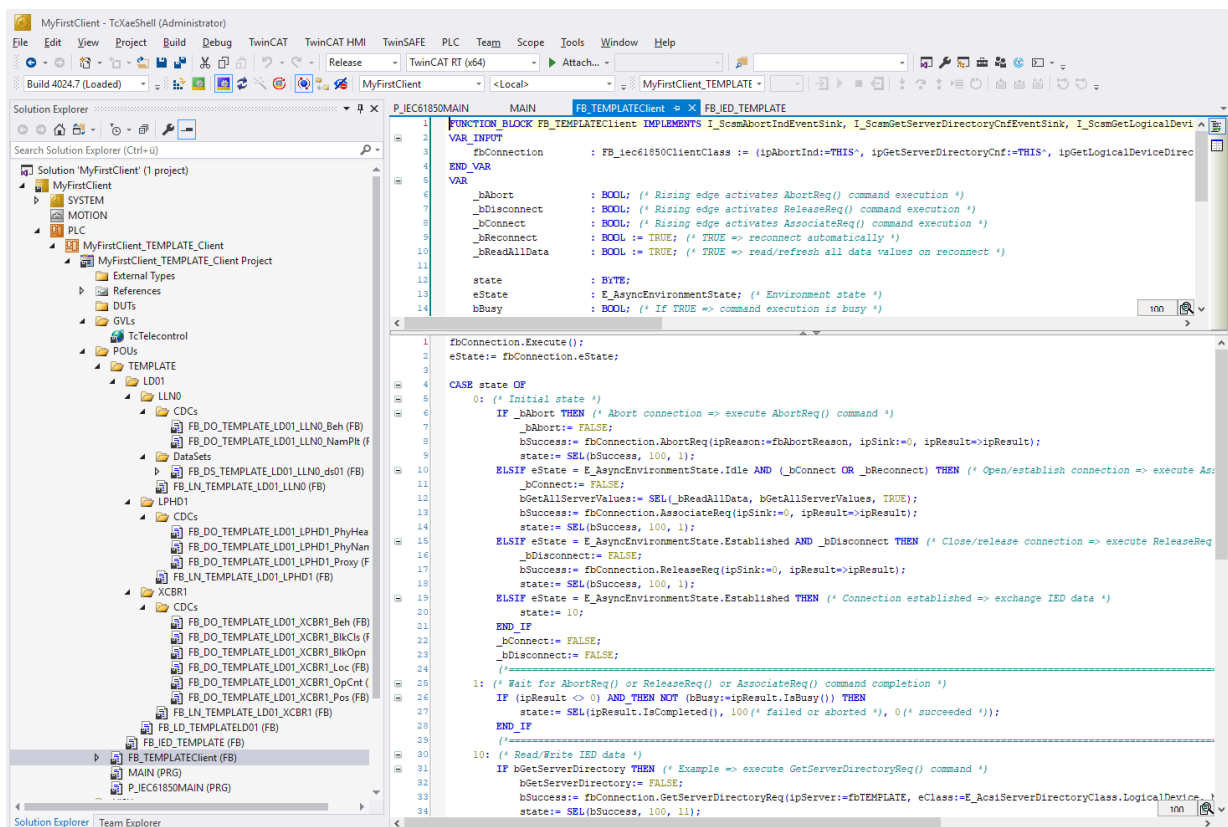


⇒ Während der SPS-Code-Generierung, wird der Fortschritt im sich automatisch öffnenden Log-Fenster angezeigt.

- Um genauere Informationen zu den einzelnen Kategorien, wie IDE oder TwinCAT PLC Code Generation zu erhalten, können Sie die jeweilige Kategorie anklicken. Mit der Bring to Front-Option bleibt das Log-Fenster dauerhaft im Vordergrund.



- ⇒ Nach der abgeschlossenen Generierung ist eine neue Instanz der ausgewählten Entwicklungsumgebung geöffnet. Der SPS-Code ist vollkommen transparent, sodass die Applikation bestmöglich geschrieben werden kann.



- ⇒ Unabhängig davon, ob SPS-Code für einen IEC 61850 Client oder Server generiert wird, bildet die globale Variablenliste TcTelecontrol den Einstiegspunkt.

```
VAR_GLOBAL
  ipCreator      : I_AcsiCodeCreatorClass := GVL_AcsiVars.Creator.SetCodeRev(codeRev:=2).SetGuiVer(major:=1, minor:=0, build:=93, revision:=10);
  fb[IEDName]   : FB_IED_[IEDName];
  fb[IEDName]Client : FB_[IEDName]Client := (fbConnection:=(ipIED:=fb[IEDName], settings:=(sRemoteHost:='127.0.0.1')));
  fb[IEDName]Gse : FB_[IEDName]Gse := (fbAdapter:=(ipIED:=fb[IEDName], settings:=(sMulticastAddr:='01-0C-CD-01-00-00', eDispatchMode:=E_GseDispatchMode.NonPromiscuous)));
END_VAR
```

Neben dem Datenmodell finden Sie dort z. B. für den Client den FB_[IEDName]Client. Dies ist ein Applikationsbaustein, der den FB_iec61850ClientClass und die komplette Kommunikation kapselt. Hier ist es besonders wichtig die IP-Adresse des Servers anzugeben, was Sie auch schon im Konfigurator tun können.

Generieren Sie einen Server, sieht die globale Variablenliste etwas anders aus.

```
VAR_GLOBAL
  ipCreator      : I_AcsiCodeCreatorClass := GVL_AcsiVars.Creator.SetCodeRev(codeRev:=2).SetGuiVer(major:=1, minor:=1, build:=94, revision:=1);
  fb[IEDName]   : FB_IED_[IEDName];
  fb[IEDName]Server : FB_iec61850ServerClass := (ipIED:=fb[IEDName], settings:=(bEnable:=TRUE, sLocalHost:='127.0.0.1'));
  fb[IEDName]ServerSession1 : FB_[IEDName]ServerSession := (fbConnection:=(ipServer:=fb[IEDName]Server, settings:=(bEnable:=TRUE)));
  fb[IEDName]Gse : FB_[IEDName]Gse := (fbAdapter:=(ipIED:=fb[IEDName], settings:=(sMulticastAddr:='01-0C-CD-01-00-00', eDispatchMode:=E_GseDispatchMode.NonPromiscuous)));
END_VAR
```

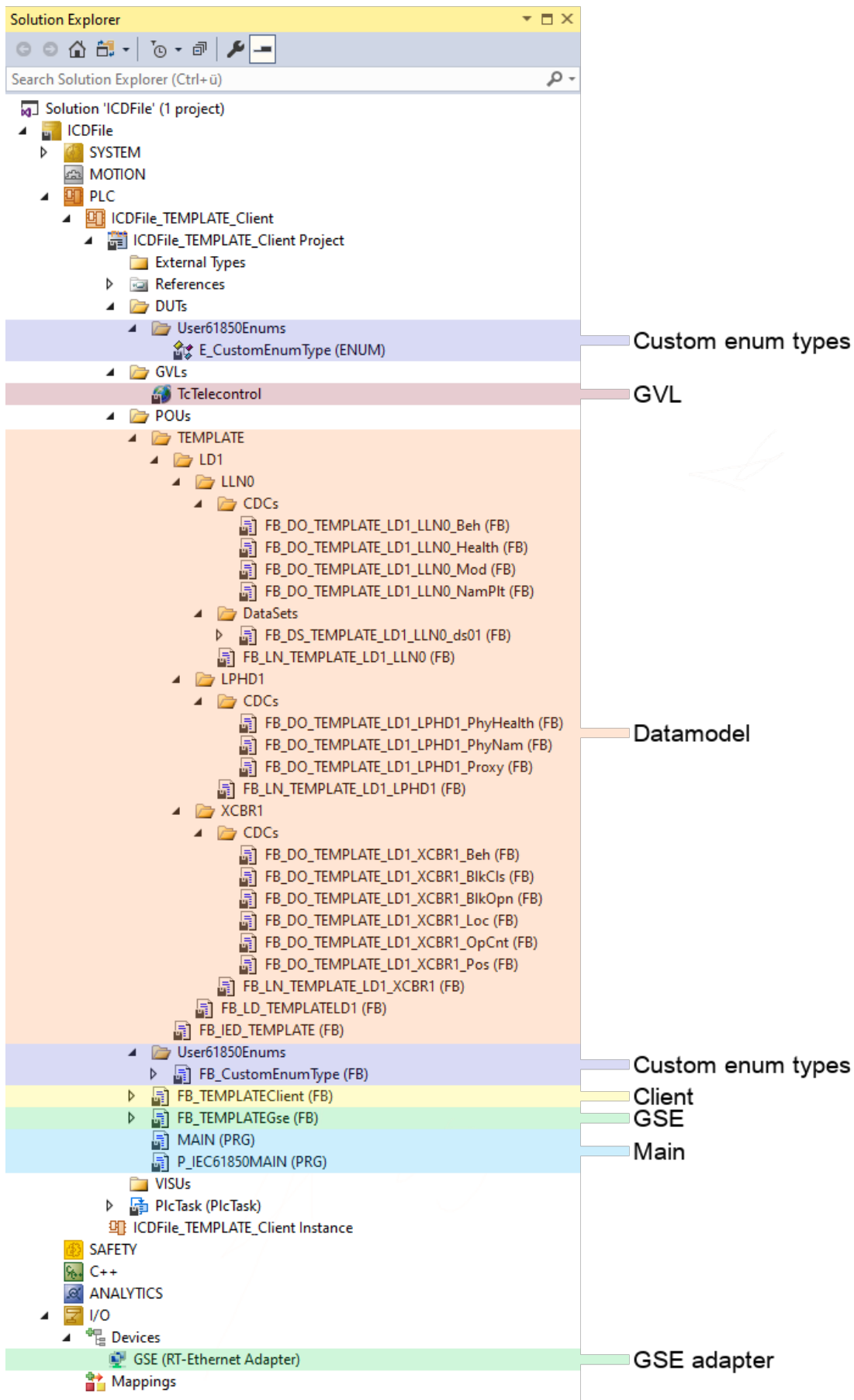
Der FB_iec61850ServerClass und die für die Kommunikation nötigen Sessions sind nicht in einem Applikationsbaustein gekapselt, da hier auch mehrere Sessions parallel für unterschiedliche Clients angelegt werden können.

5.8.1 Codeversion 2.0

Die Codeversion wird seit dem Release v1.1.94.2 ausgeliefert. Sie ist auch immer in dem generierten SPS Code in den Globalen Variablen TcTelecontrol aufgeführt.

```
ipCreator :I_AcsiCodeCreatorClass := GVL_AcsiVars.Creator.SetCodeRev(codeRev:=2).SetGuiVer(major:=1,  
minor:=1, build:=94, revision:=2);
```

5.8.1.1 Client



Custom enum types:

Benutzerdefinierte, nicht im Standard vorhandene Enumtypen.

GVL:

In der globalen Variablenliste werden die Client-Verbindungsinstanz, die IED-Datenmodell-Bausteininstanz und optional ein oder mehrere GOOSE-Kommunikations-Bausteininstanzen instanziiert und initialisiert.

Außerdem wird dort auch die, während der Codegenerierung verwendete Codeversion und die verwendete Version des TwinCAT Telecontrol Configurator abgelegt.

Datamodel:

Die gesamte hierarchische Struktur des IED-Datenmodells als Funktionsbausteine. Beinhaltet auch den IED-Baustein, der in der globalen Variablenliste instanziiert wird.

Client:

Dieser Baustein beinhaltet eine State-Machine, von der die grundlegenden Zustände in jedem Client-Beispiel genutzt werden.

Die Bausteininstanz wird für das Verbindungsmanagement und den Datenaustausch benötigt.

GSE / GSE Adapter:

Dieser Baustein verbindet einen Netzwerkadapter des TwinCAT-Steuerrechners mit den GOOSE-Kontrollblöcken im Datenmodell und steuert die Ausführung dieser Kontrollblöcke.

Um die GOOSE-Kommunikation nutzen zu können, muss der Echtzeit-Netzwerkadapter in Betrieb genommen und verknüpft werden. Dies ist in [RT - Ethernet Adapter Konfiguration \[► 37\]](#) erläutert.

Beim GSE-Adapter handelt es sich um einen Netzwerkadapter für die GOOSE Kommunikation, dessen Netzwerkstatus ständig vom GSE-Funktionsbaustein abgefragt wird.

Main:

Das TwinCAT SPS-Projekt enthält standardmäßig ein „MAIN“-Programm. Dieses wird zyklisch von einer TwinCAT Task aufgerufen und ruft wiederum das Programm „P_IEC61850MAIN“ auf.

Das Programm „P_IEC61850MAIN“ kapselt den Aufruf des Client-Bausteins und des optionalen GSE-Bausteins, trennt die IEC 61850-Kommunikation von dem restlichen SPS-Maschinenprogramm und hilft z. B. bei der Implementierung weiterer Clients.

5.8.1.1.1 E_[EnumName]

Benutzerdefinierte, nicht im Standard vorhandene Enumtypen.

Die maximalen Indizes sind -32768 und 32767, da es sich um einen 16bit Integer handelt. Die Benennung der Enum-Werte aus der ICD-Datei wird lediglich als Kommentar übernommen.

Syntax

Beispiel-Definition:

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_[EnumName] :
(
  e_1 := 1, (*Ok*)
  e_2 := 2, (*Warning*)
  e_3 := 3, (*Alarm*)
  ...
) SINT;
END_TYPE
```

Werte

Name	Beschreibung
e_1	Benutzerdefinierter Enum-Wert, entspricht „Ok“ in der ICD-Datei.
e_2	Benutzerdefinierter Enum-Wert, entspricht „Warning“ in der ICD-Datei.
e_3	Benutzerdefinierter Enum-Wert, entspricht „Alarm“ in der ICD-Datei.
...	Beliebige weitere, benutzerdefinierte Enum-Werte im Indexbereich zwischen -32768 und 32767.

5.8.1.1.2 TcTelecontrol (GVL)

In der globalen Variablenliste werden die Client-Verbindungsinstanz, die IED-Datenmodell-Bausteininstanz und optional ein oder mehrere GOOSE-Kommunikations-Bausteininstanzen instanziiert und initialisiert.

Außerdem wird dort auch die, während der Codegenerierung verwendete Codeversion und die verwendete Version des TwinCAT Telecontrol Configurator abgelegt.

[Vollständige Beschreibung \[► 27\]](#)

Syntax

Definition:

```
VAR_GLOBAL
    ipCreator      : I_AcsiCodeCreatorClass := GVL_AcsiVars.Creator.SetCodeRev(codeRev:=2).SetGui
Ver(major:=1, minor:=0, build:=93, revision:=10);
    fb[IEDName]   : FB_IED_[IEDName];
    fb[IEDName]Client : FB_[IEDName]Client := (fbConnection:=(ipIED:=fb[IEDName], settings:=(sRemoteH
ost:='127.0.0.1')));
    fb[IEDName]Gse : FB_[IEDName]Gse := (fbAdapter:=(ipIED:=fb[IEDName], settings:=(sMulticastAdd
r:='01-0C-CD-01-00-00', eDispatchMode:=E_GseDispatchMode.NonPromiscuous)));
END_VAR
```

5.8.1.1.3 FB_[EnumName]

Benutzerdefinierte, nicht im Standard vorhandene Enumtypen.

[Vollständige Beschreibung \[► 98\]](#)

Syntax

Definition:

```
{attribute 'no_explicit_call' := 'Direct call not allowed for type FB_[EnumName]'}
{attribute 'call_after_init'}
FUNCTION_BLOCK FB_[EnumName] EXTENDS FB_AcsiBTypeENUMERATED
VAR
    _eValue: E_CustomEnumType;
    _eConfig: E_CustomEnumType;
    {attribute 'hide'}
    ipInit: I_AcsiCommonAttributeClass := THIS^.AddDAValue(eP:=E_AcsiP.Actual, in:=_eValue).AddDAVal
ue(eP:=E_AcsiP.Config, in:=_eConfig).AddDAType(in:=THIS^.T(in:=E_AcsiBasicType.Enum));
END_VAR
```

5.8.1.1.4 FB_[IEDName]Client

Dieser Baustein beinhaltet eine State-Machine, von der die grundlegenden Zustände in jedem Client-Beispiel genutzt werden.

Die Bausteininstanz wird für das Verbindungsmanagement und den Datenaustausch benötigt.

[Vollständige Beschreibung \[► 27\]](#)

Syntax

Deklaration:

```

FUNCTION_BLOCK FB_[IEDName]Client
VAR_INPUT
    fbConnection          : FB_iec61850ClientClass;
END_VAR
VAR
    _bAbort              : BOOL;
    _bDisconnect         : BOOL;
    _bConnect            : BOOL;
    _bReconnect          : BOOL := TRUE;
    _bReadAllData        : BOOL := TRUE;
    state                : BYTE;
    eState               : E_AsyncEnvironmentState;
    bBusy               : BOOL;
    bSuccess             : BOOL;
    ipResult             : I_AsyncServiceResultClass;
    sLastErrorResult     : T_MaxString;
    fbAbortReason        : FB_ServiceErrorClass := (stError:=SUCCESS_EVENT);
    sLastAbortReason     : T_MaxString;
    nInvokeID           : UDINT;
    eServiceError        : E_AcsiServiceError;
    nServiceError        : UDINT;
    nCmdError            : UDINT;
    sObjReference        : T_AcsiObjectReference;
    sCtrlReference       : T_AcsiObjectReference;

    bGetServerDirectory : BOOL := TRUE;
    bGetLogicalDeviceDirectory: BOOL := TRUE;
    bGetLogicalNodeDirectory : BOOL := TRUE;
    bGetAllServerValues  : BOOL := TRUE;
    bGetAllServerValues  : BOOL := TRUE;
    bGetAllDataValues_LLNO_ST : BOOL := TRUE;
    bGetDataValues_LLNO_ST_Beh: BOOL := TRUE;
END_VAR

fbConnection.Execute();
eState:= fbConnection.eState;

CASE state OF
    0:
        IF _bAbort THEN
            _bAbort:= FALSE;
            bSuccess:= fbConnection.AbortReq(ipReason:=fbAbortReason, ipSink:=0, ipResult=>ipResult)
;
            state:= SEL(bSuccess, 100, 1);
        ELSIF eState = E_AsyncEnvironmentState.Idle AND (_bConnect OR _bReconnect) THEN
            _bConnect:= FALSE;
            bGetAllServerValues:= SEL(_bReadAllData, bGetAllServerValues, TRUE);
            bSuccess:= fbConnection.AssociateReq(ipSink:=0, ipResult=>ipResult);
            state:= SEL(bSuccess, 100, 1);
        ELSIF eState = E_AsyncEnvironmentState.Established AND _bDisconnect THEN
            _bDisconnect:= FALSE;
            bSuccess:= fbConnection.ReleaseReq(ipSink:=0, ipResult=>ipResult);
            state:= SEL(bSuccess, 100, 1);
        ELSIF eState = E_AsyncEnvironmentState.Established THEN
            state:= 10;
        END_IF
        _bConnect:= FALSE;
        _bDisconnect:= FALSE;
    1:
        IF ipResult <> 0 THEN
            ipResult.Execute();
            IF NOT (bBusy:=ipResult.IsBusy()) THEN
                state:= SEL(ipResult.IsCompleted(), 100, 0);
            END_IF
        END_IF
    10:
        IF bGetServerDirectory THEN
            bGetServerDirectory:= FALSE;
            bSuccess:= fbConnection.GetServerDirectoryReq(ipServer:=fb[IEDName], eClass:=E_AcsiServerDirectoryClass.LogicalDevice, hUser:=0, ipSink:=0, nInvokeID=>nInvokeID, ipResult=>ipResult);
            state:= SEL(bSuccess, 100, 11);
        ELSIF bGetLogicalDeviceDirectory THEN
            bGetLogicalDeviceDirectory:= FALSE;
            bSuccess:= fbConnection.GetLogicalDeviceDirectoryReq(ipLogicalDevice:=fb[IEDName].IEDLD1, hUser:=0, ipSink:=0, nInvokeID=>nInvokeID, ipResult=>ipResult);
            state:= SEL(bSuccess, 100, 11);

```

```

        ELSIF bGetLogicalNodeDirectory THEN
            bGetLogicalNodeDirectory:= FALSE;
            bSuccess:= fbConnection.GetLogicalNodeDirectoryReq(ipLogicalNode:=fb[IEDName].IEDLD1.LLN
0, eClass:=E_AcsiLogicalNodeClass.DataSet, hUser:=0, ipSink:=0, nInvokeID=>nInvokeID, ipResult=>ipRe
sult);
            state:= SEL(bSuccess, 100, 11);
        ELSIF bGetAllServerValues THEN
            bGetAllServerValues:= FALSE;
            bSuccess:= fbConnection.GetAllServerValuesReq(ipServer:=fb[IEDName], hUser:=0, ipSink:=0
, nInvokeID=>nInvokeID, ipResult=>ipResult);
            state:= SEL(bSuccess, 100, 11);
        ELSIF bGetAllDataValues_LLNO_ST THEN
            bGetAllDataValues_LLNO_ST:= FALSE;
            bSuccess:= fbConnection.GetAllDataValuesReq(ipLogicalNode:=fb[IEDName].IEDLD1.LLN0, eFc:
=E_AcsiFc.ST_, hUser:=0, ipSink:=0, nInvokeID=>nInvokeID, ipResult=>ipResult);
            state:= SEL(bSuccess, 100, 11);
        ELSIF bGetDataValues_LLNO_ST_Beh THEN
            bGetDataValues_LLNO_ST_Beh:= FALSE;
            bSuccess:= fbConnection.GetDataValuesReq(ipData:=fb[IEDName].IEDLD1.LLN0.Beh, eFc:=E_Acs
iFc.ST_, hUser:=0, ipSink:=0, nInvokeID=>nInvokeID, ipResult=>ipResult);
            state:= SEL(bSuccess, 100, 11);
        ELSE
            state:= 0;
        END_IF
11:
    IF ipResult <> 0 THEN
        ipResult.Execute();
        IF NOT (bBusy:=ipResult.IsBusy()) THEN
            state:= SEL(ipResult.IsCompleted(), 100, 0);
        END_IF
    END_IF
100:
    state:= 0;
    IF ipResult <> 0 THEN
        nCmdError:= nCmdError + 1;
        sLastErrorResult:= ipResult.Dump();
    END_IF
END_CASE

```

5.8.1.1.5 FB_[IEDName]Gse

Dieser Baustein verbindet einen Netzwerkadapter des TwinCAT-Steuerungsrechners mit den GOOSE-Kontrollblöcken im Datenmodell und steuert die Ausführung dieser Kontrollblöcke.

Um die GOOSE-Kommunikation nutzen zu können, muss der Echtzeit-Netzwerkadapter in Betrieb genommen und verknüpft werden. Dies ist in [RT - Ethernet Adapter Konfiguration \[► 37\]](#) erläutert.

[Vollständige Beschreibung \[► 27\]](#)

Syntax

Definition:

```

FUNCTION_BLOCK FB_[IEDName]Gse IMPLEMENTS I_GseLinkStatusEventSink
VAR_INPUT
    fbAdapter      : FB_GseAdapterClass := (ipLinkStatus:=THIS^);
END_VAR
VAR
    eLinkStatus   : E_GseLinkStatus;
    bSuccess      : BOOL;
    ipError       : I_ServiceErrorClass;
    bSubscribe    : BOOL := TRUE;
    bUnsubscribe  : BOOL;
END_VAR
bSuccess:= fbAdapter.Execute(ipError=>ipError);
IF bSubscribe THEN
    bSubscribe:= FALSE;
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb01.Subscriber.Enable(ipAdapter:=fbAdapter, ipError=>ipErr
or);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb02.Subscriber.Enable(ipAdapter:=fbAdapter, ipError=>ipErr
or);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb03.Subscriber.Enable(ipAdapter:=fbAdapter, ipError=>ipErr
or);
ELSEIF bUnsubscribe THEN
    bUnsubscribe:= FALSE;
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb01.Subscriber.Disable(ipError=>ipError);

```



```
bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb02.Subscriber.Disable(ipError=>ipError);
bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb03.Subscriber.Disable(ipError=>ipError);
ELSE
bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb01.Subscriber.Execute(ipError=>ipError);
bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb02.Subscriber.Execute(ipError=>ipError);
bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb03.Subscriber.Execute(ipError=>ipError);
END_IF
```

5.8.1.1.6 MAIN

Das TwinCAT SPS-Projekt enthält standardmäßig ein „MAIN“-Programm. Dieses wird zyklisch von einer TwinCAT Task aufgerufen und ruft wiederum das Programm „P_IEC61850MAIN“ auf.

[Vollständige Beschreibung \[► 27\]](#)

Syntax

Definition:

```
PROGRAM MAIN
VAR
END_VAR

P_IEC61850MAIN();
```

5.8.1.1.7 P_IEC61850MAIN

Das Programm „P_IEC61850MAIN“ kapselt den Aufruf des Client-Bausteins und des optionalen GSE-Bausteins, trennt die IEC 61850-Kommunikation von dem restlichen SPS-Maschinenprogramm und hilft z. B. bei der Implementierung weiterer Clients.

[Vollständige Beschreibung \[► 27\]](#)

Syntax

Definition:

```
PROGRAM P_IEC61850MAIN
VAR
END_VAR

fb[IEDName]Client();
fb[IEDName]Gse();
```

5.8.1.2 Server

The screenshot shows the Solution Explorer for a project named 'ICDFile'. The structure is as follows:

- Solution 'ICDFile' (1 project)
 - ICDFile
 - SYSTEM
 - MOTION
 - PLC
 - ICDFile_TEMPLATE_Server
 - ICDFile_TEMPLATE_Server Project
 - External Types
 - References
 - DUTs
 - User61850Enums (Custom enum types)
 - E_CustomEnumType (ENUM) (GVL)
 - GVLs (GVL)
 - TcTelecontrol
 - POUs
 - TEMPLATE (Datamodel)
 - LD1
 - LLN0
 - CDCs
 - FB_DO_TEMPLATE_LD1_LLN0_Beh (FB)
 - FB_DO_TEMPLATE_LD1_LLN0_Health (FB)
 - FB_DO_TEMPLATE_LD1_LLN0_Mod (FB)
 - FB_DO_TEMPLATE_LD1_LLN0_NamPlt (FB)
 - DataSets
 - FB_DS_TEMPLATE_LD1_LLN0_ds01 (FB)
 - FB_LN_TEMPLATE_LD1_LLN0 (FB)
 - LPHD1
 - CDCs
 - FB_DO_TEMPLATE_LD1_LPHD1_PhyHealth (FB)
 - FB_DO_TEMPLATE_LD1_LPHD1_PhyNam (FB)
 - FB_DO_TEMPLATE_LD1_LPHD1_Proxy (FB)
 - FB_LN_TEMPLATE_LD1_LPHD1 (FB)
 - XCBR1
 - CDCs
 - FB_DO_TEMPLATE_LD1_XCBR1_Beh (FB)
 - FB_DO_TEMPLATE_LD1_XCBR1_BlkCls (FB)
 - FB_DO_TEMPLATE_LD1_XCBR1_BlkOpn (FB)
 - FB_DO_TEMPLATE_LD1_XCBR1_Loc (FB)
 - FB_DO_TEMPLATE_LD1_XCBR1_OpCnt (FB)
 - FB_DO_TEMPLATE_LD1_XCBR1_Pos (FB)
 - FB_LN_TEMPLATE_LD1_XCBR1 (FB)
 - FB_LD_TEMPLATELD1 (FB)
 - User61850Enums (Custom enum types)
 - FB_CustomEnumType (FB)
 - FB_TEMPLATEGse (FB) (GSE)
 - FB_TEMPLATEServerSession (FB) (Server)
 - MAIN (PRG) (Main)
 - P_IEC61850MAIN (PRG) (Main)
 - VISUs
 - PlcTask (PlcTask)
 - ICDFile_TEMPLATE_Server Instance
 - SAFETY
 - C++
 - ANALYTICS
 - I/O
 - Devices
 - GSE (RT-Ethernet Adapter) (GSE adapter)
 - Mappings

Custom enum types:

Benutzerdefinierte, nicht im Standard vorhandene Enumtypen.

GVL:

In der globalen Variablenliste werden mindestens eine Instanz des Server-Session-Bausteins, die IED-Datenmodell-Bausteininstanz und optional ein oder mehrere GOOSE-Kommunikations-Bausteininstanzen instanziiert und initialisiert.

Außerdem wird dort auch die, während der Codegenerierung verwendete Codeversion und die verwendete Version des TwinCAT Telecontrol Configurator abgelegt.

Datamodel:

Die gesamte hierarchische Struktur des IED-Datenmodells als Funktionsbausteine. Beinhaltet auch den IED-Baustein, der in der globalen Variablenliste instanziiert wird.

GSE / GSE adapter:

Dieser Baustein verbindet einen Netzwerkadapter des TwinCAT-Steuerrechners mit den GOOSE-Kontrollblöcken im Datenmodell und steuert die Ausführung dieser Kontrollblöcke.

Um die GOOSE-Kommunikation nutzen zu können, muss der Echtzeit-Netzwerkadapter in Betrieb genommen und verknüpft werden. Dies ist in [RT - Ethernet Adapter Konfiguration \[► 37\]](#) erläutert.

Beim GSE-Adapter handelt es sich um einen Netzwerkadapter für die GOOSE Kommunikation, dessen Netzwerkstatus ständig vom GSE-Funktionsbaustein abgefragt wird.

Server:

Dieser Baustein beinhaltet eine State-Machine, von der die grundlegenden Zustände in jedem Server-Beispiel genutzt werden.

Der Server-Session-Baustein ist verantwortlich für den Verbindungsaufbau und Datenaustausch zu einem einzelnen Client. Bei mehreren gleichzeitigen Clientverbindungen werden mehrere Instanzen von diesem Baustein benötigt und instanziiert.

Main:

Das TwinCAT SPS-Projekt enthält standardmäßig ein „MAIN“-Programm. Dieses wird zyklisch von einer TwinCAT Task aufgerufen und ruft wiederum das Programm „P_IEC61850MAIN“ auf.

Das Programm „P_IEC61850MAIN“ kapselt den Aufruf des Server-Session-Bausteins und des optionalen GSE-Bausteins. Außerdem trennt es die IEC 61850-Kommunikation von dem restlichen SPS-Maschinenprogramm.

5.8.1.2.1 E_[EnumName]

Benutzerdefinierte, nicht im Standard vorhandene Enumtypen.

Die maximalen Indizes sind -32768 und 32767, da es sich um einen 16bit Integer handelt. Die Benennung der Enum-Werte aus der ICD-Datei wird lediglich als Kommentar übernommen.

Syntax

Beispiel-Definition:

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_[EnumName] :
(
  e_1 := 1, (*Ok*)
  e_2 := 2, (*Warning*)
  e_3 := 3, (*Alarm*)
  ...
)
```

```
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|------|---|
| e_1 | Benutzerdefinierter Enum-Wert, entspricht „Ok“ in der ICD-Datei. |
| e_2 | Benutzerdefinierter Enum-Wert, entspricht „Warning“ in der ICD-Datei. |
| e_3 | Benutzerdefinierter Enum-Wert, entspricht „Alarm“ in der ICD-Datei. |
| ... | Beliebige weitere, benutzerdefinierte Enum-Werte im Indexbereich zwischen -32768 und 32767. |

5.8.1.2.2 TcTelecontrol (GVL)

In der globalen Variablenliste werden mindestens eine Instanz des Server-Session-Bausteins, die IED-Datenmodell-Bausteininstanz und optional ein oder mehrere GOOSE-Kommunikations-Bausteininstanzen instanziiert und initialisiert.

Außerdem wird dort auch die, während der Codegenerierung verwendete Codeversion und die verwendete Version des TwinCAT Telecontrol Configurator abgelegt.

[Vollständige Beschreibung](#) [▶ 32](#)

Syntax

Definition:

```
VAR_GLOBAL
    ipCreator          : I_AcsiCodeCreatorClass := GVL_AcsiVars.Creator.SetCodeRev(codeRev:=2
) .SetGuiVer(major:=1, minor:=1, build:=94, revision:=1);
    fb[IEDName]       : FB_IED_[IEDName];
    fb[IEDName]Server : FB_iec61850ServerClass := (ipIED:=fb[IEDName], settings:=(bEnable:=T
RUE, sLocalHost:='127.0.0.1'));
    fb[IEDName]ServerSession1 : FB_[IEDName]ServerSession := (fbConnection:=(ipServer:=fb[IEDName]Se
rver, settings:=(bEnable:=TRUE)));
    fb[IEDName]Gse     : FB_[IEDName]Gse := (fbAdapter:=(ipIED:=fb[IEDName], settings:=(sMult
icastAddr:='01-0C-CD-01-00-00', eDispatchMode:=E_GseDispatchMode.NonPromiscuous)));
END_VAR
```

5.8.1.2.3 FB_[EnumName]

Benutzerdefinierte, nicht im Standard vorhandene Enumtypen.

[Vollständige Beschreibung](#) [▶ 98](#)

Syntax

Definition:

```
{attribute 'no_explicit_call' := 'Direct call not allowed for type FB_[EnumName]'}
{attribute 'call_after_init'}
FUNCTION_BLOCK FB_[EnumName] EXTENDS FB_AcsiBATypeENUMERATED
VAR
    _eValue: E_CustomEnumType;
    _eConfig: E_CustomEnumType;
    {attribute 'hide'}
    ipInit: I_AcsiCommonAttributeClass := THIS^.AddDAValue(eP:=E_AcsiP.Actual, in:=_eValue).AddDAVal
ue(eP:=E_AcsiP.Config, in:=_eConfig).AddDAType(in:=THIS^.T(in:=E_AcsiBasicType.Enum));
END_VAR
```

5.8.1.2.4 FB_[IEDName]Gse

Dieser Baustein verbindet einen Netzwerkadapter des TwinCAT-Steuerrechners mit den GOOSE-Kontrollblöcken im Datenmodell und steuert die Ausführung dieser Kontrollblöcke.

Um die GOOSE-Kommunikation nutzen zu können, muss der Echtzeit-Netzwerkadapter in Betrieb genommen und verknüpft werden. Dies ist in [RT - Ethernet Adapter Konfiguration \[► 37\]](#) erläutert.

[Vollständige Beschreibung \[► 32\]](#)

Syntax

Definition:

```
FUNCTION_BLOCK FB_[IEDName]Gse IMPLEMENTS I_GseLinkStatusEventSink
VAR_INPUT
    fbAdapter    : FB_GseAdapterClass := (ipLinkStatus:=THIS^);
END_VAR
VAR
    eLinkStatus : E_GseLinkStatus;
    bSuccess    : BOOL;
    ipError     : I_ServiceErrorClass;
    bStart      : BOOL := TRUE;
    bStop       : BOOL;
    bUpdate     : BOOL;
END_VAR

bSuccess:= fbAdapter.Execute(ipError=>ipError);

IF bStart THEN
    bStart:= FALSE;
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb01.Publisher.Start(ipAdapter:=fbAdapter, ipError=>ipError);
);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb02.Publisher.Start(ipAdapter:=fbAdapter, ipError=>ipError);
);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb03.Publisher.Start(ipAdapter:=fbAdapter, ipError=>ipError);
);
ELSIF bStop THEN
    bStop:= FALSE;
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb01.Publisher.Stop(ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb02.Publisher.Stop(ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb03.Publisher.Stop(ipError=>ipError);
);
ELSIF bUpdate THEN
    bUpdate:= FALSE;
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb01.Publisher.Update(ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb02.Publisher.Update(ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb03.Publisher.Update(ipError=>ipError);
);
ELSE
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb01.Publisher.Execute(ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb02.Publisher.Execute(ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb03.Publisher.Execute(ipError=>ipError);
);
END_IF
```

Der GSE-Baustein implementiert die „I_GseLinkStatusEventSink“-Schnittstelle. Die Methode: „OnLinkStatusChange“ gehört zu dieser Schnittstellenimplementierung und wird immer dann aufgerufen, wenn sich der Status der Netzwerkverbindung (am Netzwerkadapter) ändert. Die SPS-Applikation kann z. B. den Netzwerkverbindungsstatus via „eLinkStatus“-Variable abfragen oder überprüfen.

```
METHOD OnLinkStatusChange
VAR_INPUT
    ipAdapter : I_GseAdapterClass;
    eStatus   : E_GseLinkStatus;
END_VAR
VAR
END_VAR
eLinkStatus:= eStatus;
```

5.8.1.2.5 FB_[IEDName]ServerSession

Dieser Baustein beinhaltet eine State-Machine, von der die grundlegenden Zustände in jedem Server-Beispiel genutzt werden.

Der Server-Session-Baustein ist verantwortlich für den Verbindungsaufbau und Datenaustausch zu einem einzelnen Client. Bei mehreren gleichzeitigen Clientverbindungen werden mehrere Instanzen von diesem Baustein benötigt und instanziiert.

[Vollständige Beschreibung \[► 32\]](#)

Syntax

Definition:

```

FUNCTION_BLOCK FB_ [IEDName]ServerSession
VAR_INPUT
    fbConnection      : FB_iec61850ConnectionClass := (ipAbortInd:=THIS^, ipAssociateInd:=THIS^, ipRe
leaseInd:=THIS^);
END_VAR
VAR
    _bAbort           : BOOL;
    _bDisconnect      : BOOL;
    state             : BYTE;
    eState            : E_AsyncEnvironmentState;
    bBusy             : BOOL;
    bSuccess          : BOOL;
    ipResult          : I_AsyncServiceResultClass;
    sLastErrorResult  : T_MaxString;
    fbAbortReason     : FB_ServiceErrorClass := (stError:=SUCCESS_EVENT);
    sLastAbortReason  : T_MaxString;
END_VAR

fbConnection.Execute();
eState:= fbConnection.eState;

CASE state OF
    0:
        IF _bAbort THEN
            _bAbort:= FALSE;
            bSuccess:= fbConnection.AbortReq(ipReason:=fbAbortReason, ipSink:=0, ipResult=>ipResult)
;
            state:= SEL(bSuccess, 100, 1);
        ELSIF eState = E_AsyncEnvironmentState.Established AND _bDisconnect THEN
            _bDisconnect:= FALSE;
            bSuccess:= fbConnection.ReleaseReq(ipSink:=0, ipResult=>ipResult);
            state:= SEL(bSuccess, 100, 1);
        ELSIF eState = E_AsyncEnvironmentState.Established THEN
            state:= 10;
        END_IF
        _bDisconnect:= FALSE;
    1:
        IF ipResult <> 0 THEN
            ipResult.Execute();
            IF NOT (bBusy:=ipResult.IsBusy()) THEN
                state:= SEL(ipResult.IsCompleted(), 100, 0);
            END_IF
        END_IF
    10:
        state:= 0;
    100:
        state:= 0;
        IF ipResult <> 0 THEN
            sLastErrorResult:= ipResult.Dump();
        END_IF
END_CASE

```

5.8.1.2.6 MAIN

Das TwinCAT SPS-Projekt enthält standardmäßig ein „MAIN“-Programm. Dieses wird zyklisch von einer TwinCAT Task aufgerufen und ruft wiederum das Programm „P_IEC61850MAIN“ auf.

[Vollständige Beschreibung \[▶ 32\]](#)

Syntax

Definition:

```

PROGRAM MAIN
VAR
END_VAR
P_IEC61850MAIN();

```

5.8.1.2.7 P_IEC61850MAIN

Das Programm „P_IEC61850MAIN“ kapselt den Aufruf des Server-Session-Bausteins und des optionalen GSE-Bausteins. Außerdem trennt es die IEC 61850-Kommunikation von dem restlichen SPS-Maschinenprogramm.

[Vollständige Beschreibung \[► 32\]](#)

Syntax

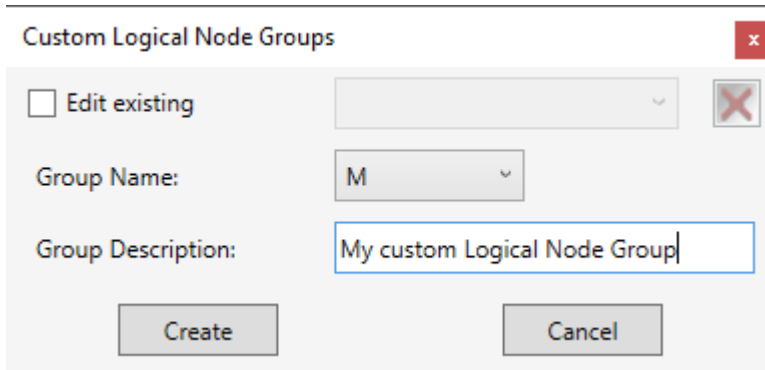
Definition:

```
PROGRAM P_IEC61850MAIN
VAR
END_VAR
```

```
fb[IEDName]Server.Execute();
fb[IEDName]ServerSession1();
fb[IEDName]Gse();
```

5.9 Benutzerdefinierte Logical Node Groups

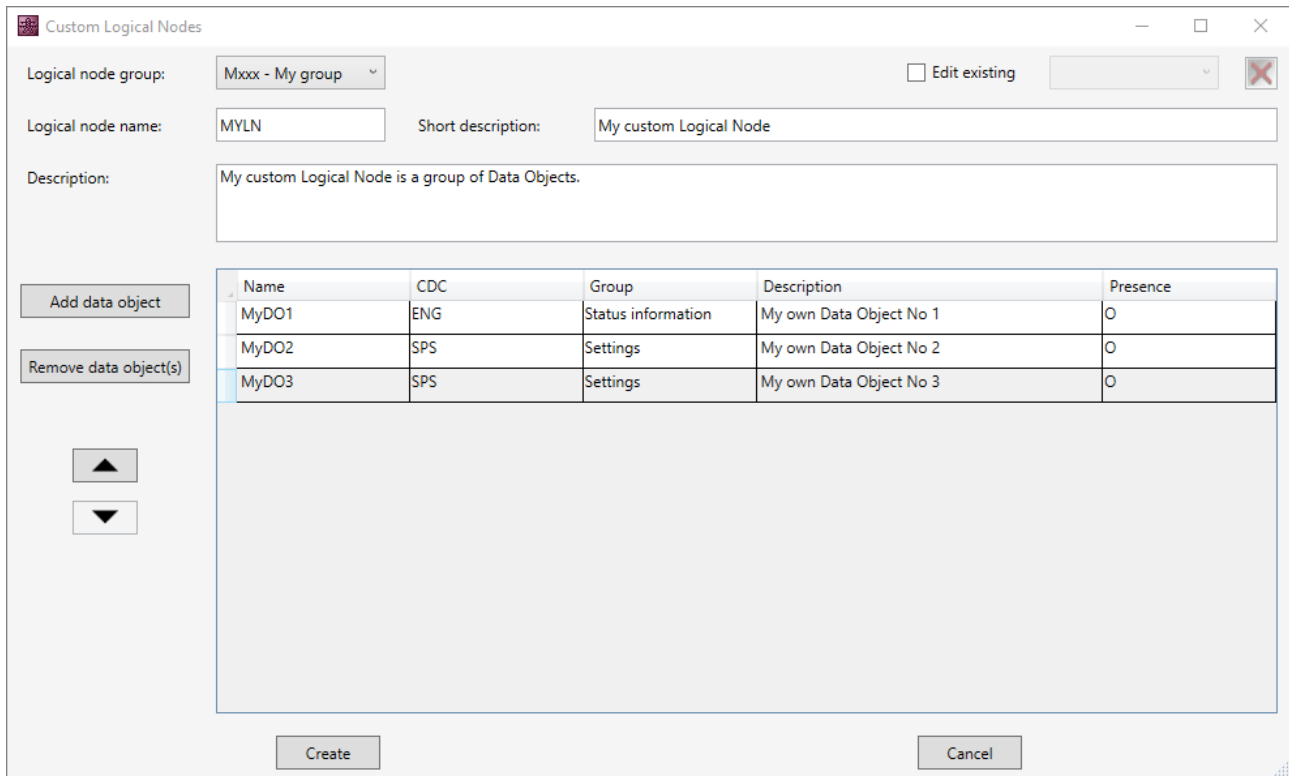
Mit der Funktion *Custom Logical Node Groups* ist es möglich eigene Logical Node Groups zu definieren. Die neuen Elemente werden dabei in eine private Datenbank geschrieben. So stehen diese auch für jede neue Konfiguration wieder zur Verfügung.



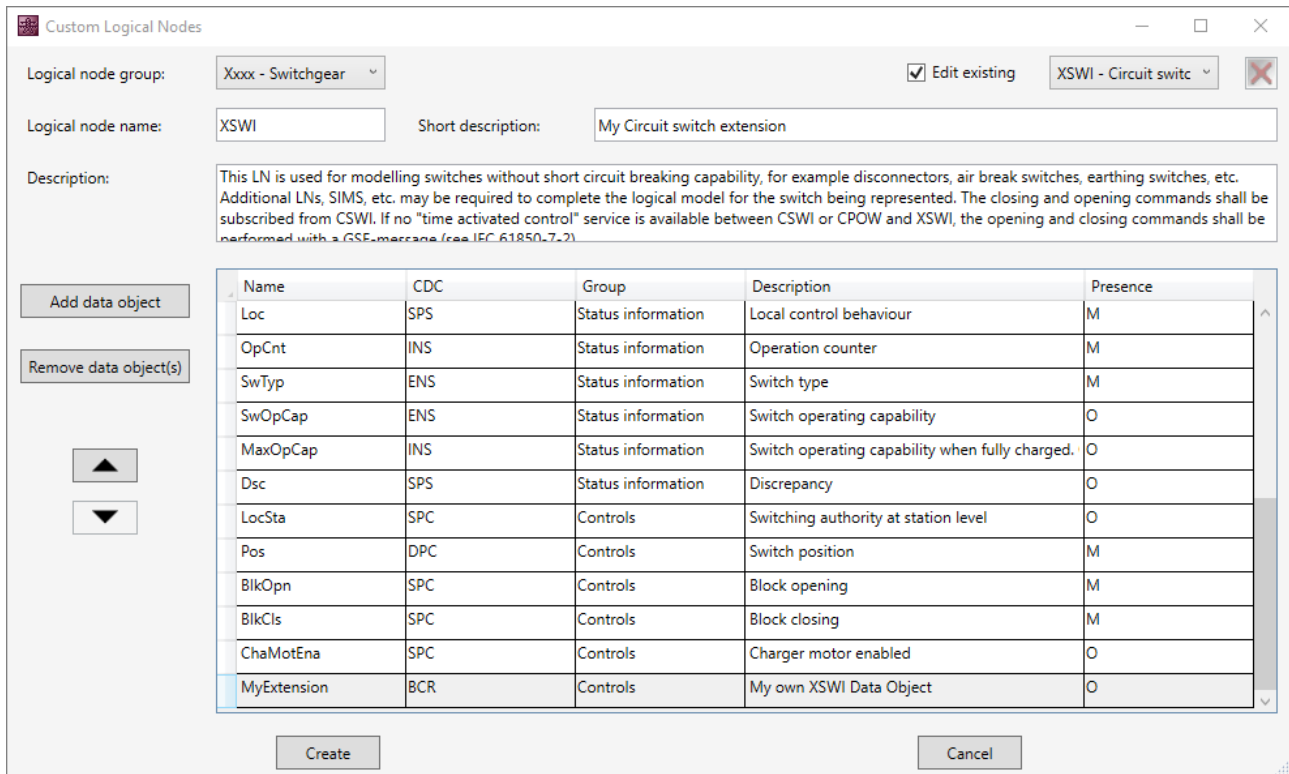
In dem gezeigten Fenster können **Name** und **Beschreibung** der Logical Node Group angepasst werden. Durch die Bestätigung mit dem Button **Create** wird die neue Logical Node Group in die private Datenbank geschrieben und steht ab sofort für die Konfiguration zur Verfügung. Oben im Fenster kann zudem eine bereits erstellte Logical Node Group ausgewählt werden, um diese zu bearbeiten oder zu löschen.

5.10 Benutzerdefinierte Logical Nodes

Mit der Funktion *Custom Logical Nodes* ist es möglich eigene Logical Nodes zu definieren oder bestehende Logical Nodes zu erweitern. Die neuen Elemente werden dabei in eine private Datenbank geschrieben. So stehen diese auch für jede neue Konfiguration wieder zur Verfügung.



In dem gezeigten Fenster können eigene Logical Nodes definiert werden. Dafür kann für jedes Data Object Name und eine Common Data Class ausgewählt werden. Des Weiteren kann das Data Object einer Gruppe zugeordnet werden, damit eine Kategorisierung im Konfigurator zu einer Steigerung der Übersichtlichkeit führt. In der Spalte Presence ist einstellbar, ob das Element standardmäßig hinzugefügt werden soll, oder es optional sein soll. Durch die Pfeile auf der linken Seite des Fensters kann die Reihenfolge der Data Objects angepasst werden. Zudem können neue Data Objects hinzugefügt oder bestehende gelöscht werden.



In der hier gezeigten Abbildung wird der im Standard definierte Logical Node **XSWI (Circuit Switch)** benutzerdefiniert erweitert. Dazu kann oben rechts im Fenster ein **Logical Node** zum Editieren ausgewählt werden. Pro Logical Node aus dem Standard kann ein benutzerdefinierter Logical Node mit demselben Namen erstellt werden. Weitere benutzerdefinierte Logical Nodes müssen individuelle Namen erhalten.

Durch die Bestätigung mit dem Button **Create** wird der neue Logical Node in die private Datenbank geschrieben und steht ab sofort für die Konfiguration zur Verfügung.

5.11 Benutzerdefinierte Common Data Classes

Mit der Funktion *Custom Common Data Classes* ist es möglich, eigene Common Data Classes zu definieren oder bestehende Common Data Classes zu erweitern. Die neuen Elemente werden dabei in eine private Datenbank geschrieben. So stehen diese auch für jede neue Konfiguration wieder zur Verfügung.

| Name | Datatype | FC | Trigger options | Presence |
|------|----------|----|-----------------|----------|
| t | INT32 | ST | dchg | M |

In dem gezeigten Fenster können eigene Common Data Classes definiert werden. Dafür kann für jedes Data Attribute ein Name, Datentyp, ein Functional Constraint und die Trigger Options eingestellt werden. In der Spalte Presence ist einstellbar, ob das Element standardmäßig hinzugefügt werden soll, oder es optional sein soll. Durch die Pfeile auf der linken Seite des Fensters kann die Reihenfolge der Data Attributes angepasst werden. Zudem können neue Data Attributes hinzugefügt oder bestehende gelöscht werden.

Common data class name: MYCDC Edit existing

Description: My custom Common Data Class

Data attributes | Sub data objects

Add sub data object

Remove sub data object(s)

| Name | CDC | Presence |
|------|-----|----------|
| t | INS | O |

▲

▼

Create Cancel

Zudem können die Sub Data Objects der Common Data Class konfiguriert werden. Für die einzelnen Sub Data Objects können Name und Common Data Class ausgewählt werden. In der Spalte Presence ist einstellbar, ob das Element standardmäßig hinzugefügt werden soll, oder es optional sein soll. Durch die Pfeile auf der linken Seite des Fensters kann die Reihenfolge der Sub Data Objects angepasst werden. Zudem können neue Sub Data Objects hinzugefügt oder bestehende gelöscht werden.

Common data class name: DEL Edit existing DEL

Description: Delta

Data attributes | Sub data objects

Add data attribute

Remove data attribute(s)

| Name | Datatype | FC | Trigger options | Presence |
|-------------|---------------------|----|-----------------|----------|
| angRef | PhaseAngleReference | CF | dchg | O |
| d | VISIBLE STRING255 | DC | none | O |
| dU | UNICODE STRING255 | DC | none | O |
| cdcNs | VISIBLE STRING255 | EX | none | O |
| cdcName | VISIBLE STRING255 | EX | none | O |
| dataNs | VISIBLE STRING255 | EX | none | O |
| myExtension | BOOLEAN | ST | none | O |

▲

▼

Create Cancel

In der hier gezeigten Abbildung wird die im Standard definierte Common Data Class **DEL (Delta)** benutzerdefiniert erweitert. Dazu kann oben rechts im Fenster eine **Common Data Class** zum Editieren ausgewählt werden. Pro Common Data Class aus dem Standard kann eine benutzerdefinierte Common Data Class mit demselben Namen erstellt werden. Weitere benutzerdefinierte Common Data Classes müssen individuelle Namen erhalten.

Durch die Bestätigung mit dem Button **Create** wird die neue Common Data Class in die private Datenbank geschrieben und steht ab sofort für die Konfiguration zur Verfügung.

6 SPS-API

6.1 Bibliotheken

TwinCAT IEC 61131 SPS Bibliotheken.

- Tc3_Acsi (Tc3_Acsi.compiled-library)
- Tc3_Ber (Tc3_Ber.compiled-library)
- Tc3_Collections (Tc3_Collections.compiled-library)
- Tc3_Gse (Tc3_Gse.compiled-library)
- Tc3_iec61850 (Tc3_iec61850.compiled-library)
- Tc3_iec61850_8_1 (Tc3_iec61850_8_1.compiled-library)
- Tc3_Mms (Tc3_Mms.compiled-library)
- Tc3_Rfc1006 (Tc3_Rfc1006.compiled-library)
- Tc3_Sockets (Tc3_Sockets.compiled-library)
- Tc3_Tpkt (Tc3_Tpkt.compiled-library)
- Tc3_Ulosi (Tc3_Ulosi.compiled-library)

6.2 Namensräume

6.2.1 TwinCAT TypeSystem Namensraum

Typ: TwinCAT Systemresource

Bibliothek: Nicht erforderlich

Globale Variablenlisten

- [TC_EVENTS](#) [► 489]
- [TC_EVENT_CLASSES](#) [► 583]

6.2.2 Tc3_Acsi Namensraum

Typ: IEC 61131 SPS Bibliothek

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Globale Variablenlisten

- [GVL_Acsi](#) [► 586]

Parameterlisten

- [Param_Acsi](#) [► 483]

6.2.3 Tc3_Ber Namensraum

Typ: IEC 61131 SPS Bibliothek

Bibliothek: Tc3_Ber (Tc3_Ber.compiled-library)

6.2.4 Tc3_Collections Namensraum

Typ: IEC 61131 SPS Bibliothek

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

6.2.5 Tc3_Gse Namensraum

Typ: IEC 61131 SPS Bibliothek

Bibliothek: Tc3_Gse (Tc3_Gse.compiled-library)

6.2.6 Tc3_iec61850_8_1 Namensraum

Typ: IEC 61131 SPS Bibliothek

Bibliothek: Tc3_iec61850_8_1 (Tc3_iec61850_8_1.compiled-library)

6.2.7 Tc3_iec61850 Namensraum

Typ: IEC 61131 SPS Bibliothek

Bibliothek: Tc3_iec61850 (Tc3_iec61850.compiled-library)

6.2.8 Tc3_Mms Namensraum

Typ: IEC 61131 SPS Bibliothek

Bibliothek: Tc3_Mms (Tc3_Mms.compiled-library)

6.2.9 Tc3_Rfc1006 Namensraum

Typ: IEC 61131 SPS Bibliothek

Bibliothek: Tc3_Rfc1006 (Tc3_Rfc1006.compiled-library)

6.2.10 Tc3_Sockets Namensraum

Typ: IEC 61131 SPS Bibliothek

Bibliothek: Tc3_Sockets (Tc3_Sockets.compiled-library)

6.2.11 Tc3_Tpkt Namensraum

Typ: IEC 61131 SPS Bibliothek

Bibliothek: Tc3_Tpkt (Tc3_Tpkt.compiled-library)

6.2.12 Tc3_Ulosi Namensraum

Typ: IEC 61131 SPS Bibliothek

Bibliothek: Tc3_Ulosi (Tc3_Ulosi.compiled-library)

6.2.13 Tc2_Tcplp Namensraum

Typ: IEC 61131 SPS Bibliothek

Bibliothek: Tc2_Tcplp (Tc2_Tcplp.compiled-library)

6.2.14 Tc2_System Namensraum

Typ: IEC 61131 SPS Bibliothek

Bibliothek: Tc2_System (Tc2_System.compiled-library)

6.3 Funktionsbausteine

6.3.1 Datenmodell

6.3.1.1 Datenattribute

6.3.1.1.1 FB_AcsiBTypeBOOLEAN

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [▶ 154] -> [FB_AcsiCommonDataClass](#) [▶ 148] -> [FB_AcsiCommonAttributeClass](#) [▶ 146] -> **FB_AcsiBTypeBOOLEAN**

```
FUNCTION_BLOCK FB_AcsiBTypeBOOLEAN EXTENDS FB_AcsiCommonAttributeClass
```



Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|------|---------|---------------|
| bValue | BOOL | Get,Set | Attributwert. |

6.3.1.1.2 FB_AcsiBTypeCODEDENUM

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [▶ 154] -> [FB_AcsiCommonDataClass](#) [▶ 148] -> [FB_AcsiCommonAttributeClass](#) [▶ 146] -> **FB_AcsiBTypeCODEDENUM**

```
FUNCTION_BLOCK FB_AcsiBTypeCODEDENUM EXTENDS FB_AcsiCommonAttributeClass
```



Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|---------|-------------|----------|---|
| bFixLen | BOOL | Get,Set | Attribut-Datenlänge (FALSE:=Variable Länge (Standardwert), TRUE:=Fixe Länge). |
| sAny | T_MaxString | Get, Set | Attributwert als formatierter Binärstring. |

6.3.1.1.3 FB_AcsiBTypeENUMERATED

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [▶ 154] -> [FB_AcsiCommonDataClass](#) [▶ 148] -> [FB_AcsiCommonAttributeClass](#) [▶ 146] -> **FB_AcsiBTypeENUMERATED**

```
FUNCTION_BLOCK FB_AcsiBTypeENUMERATED EXTENDS FB_AcsiCommonAttributeClass
```

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|------|----------|--|
| iAny | SINT | Get,Set | Attributwert als 8 Bit Zahl mit Vorzeichen. |
| i16Any | INT | Get, Set | Attributwert als 16 Bit Zahl mit Vorzeichen. |

6.3.1.1.4 FB_AcsiBTypeFLOAT32

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> **FB_AcsiBTypeFLOAT32**

FUNCTION_BLOCK FB_AcsiBTypeFLOAT32 EXTENDS FB_AcsiCommonAttributeClass

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|------|---------|--|
| fValue | REAL | Get,Set | Attributwert, 32 Bit Fließkommazahl mit einer einfachen Genauigkeit nach IEEE 754. |

6.3.1.1.5 FB_AcsiBTypeFLOAT64

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> **FB_AcsiBTypeFLOAT64**

FUNCTION_BLOCK FB_AcsiBTypeFLOAT64 EXTENDS FB_AcsiCommonAttributeClass

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|-------|---------|--|
| fValue | LREAL | Get,Set | Attributwert, 64 Bit Fließkommazahl mit einer doppelten Genauigkeit nach IEEE 754. |

6.3.1.1.6 FB_AcsiBTypeINT8

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> **FB_AcsiBTypeINT8**

FUNCTION_BLOCK FB_AcsiBTypeINT8 EXTENDS FB_AcsiCommonAttributeClass

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|------|---------|---------------|
| iValue | SINT | Get,Set | Attributwert. |

6.3.1.1.7 FB_AcsiBATypeINT8U

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> **FB_AcsiBATypeINT8U**

```
FUNCTION_BLOCK FB_AcsiBATypeINT8U EXTENDS FB_AcsiCommonAttributeClass
```

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|------|---------|---------------|
| nValue | BYTE | Get,Set | Attributwert. |

6.3.1.1.8 FB_AcsiBATypeINT16

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> **FB_AcsiBATypeINT16**

```
FUNCTION_BLOCK FB_AcsiBATypeINT16 EXTENDS FB_AcsiCommonAttributeClass
```

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|-----|---------|---------------|
| iValue | INT | Get,Set | Attributwert. |

6.3.1.1.9 FB_AcsiBATypeINT16U

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> **FB_AcsiBATypeINT16U**

```
FUNCTION_BLOCK FB_AcsiBATypeINT16U EXTENDS FB_AcsiCommonAttributeClass
```

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|------|---------|---------------|
| nValue | WORD | Get,Set | Attributwert. |

6.3.1.1.10 FB_AcsiBATypeINT24

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> **FB_AcsiBATypeINT24**

FUNCTION_BLOCK FB_AcsiBATypeINT24 EXTENDS FB_AcsiCommonAttributeClass

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|-----------------------------------|---------|---------------|
| iValue | T_INT24 [▶ 477] | Get,Set | Attributwert. |

6.3.1.1.11 FB_AcsiBATypeINT24U

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> **FB_AcsiBATypeINT24U**

FUNCTION_BLOCK FB_AcsiBATypeINT24U EXTENDS FB_AcsiCommonAttributeClass

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|------------------------------------|---------|---------------|
| nValue | T_UINT24 [▶ 479] | Get,Set | Attributwert. |

6.3.1.1.12 FB_AcsiBATypeINT32

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> **FB_AcsiBATypeINT32**

FUNCTION_BLOCK FB_AcsiBATypeINT32 EXTENDS FB_AcsiCommonAttributeClass

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|------|---------|---------------|
| iValue | DINT | Get,Set | Attributwert. |

6.3.1.1.13 FB_AcsiBATypeINT32U

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> **FB_AcsiBATypeINT32U**

FUNCTION_BLOCK FB_AcsiBATypeINT32U EXTENDS FB_AcsiCommonAttributeClass

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|-------|---------|---------------|
| nValue | DWORD | Get,Set | Attributwert. |

6.3.1.1.14 FB_AcsiBATypeINT64

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeINT64](#)

```
FUNCTION_BLOCK FB_AcsiBATypeINT64 EXTENDS FB_AcsiCommonAttributeClass
```

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|------|---------|---------------|
| iValue | LINT | Get,Set | Attributwert. |

6.3.1.1.15 FB_AcsiBATypeINT64U

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeINT64U](#)

```
FUNCTION_BLOCK FB_AcsiBATypeINT64U EXTENDS FB_AcsiCommonAttributeClass
```

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|-------|---------|---------------|
| nValue | LWORD | Get,Set | Attributwert. |

6.3.1.1.16 FB_AcsiBATypeINT128

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeINT128](#)

```
FUNCTION_BLOCK FB_AcsiBATypeINT128 EXTENDS FB_AcsiCommonAttributeClass
```

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|---------------------------------|---------|---------------|
| iValue | T_INT28 [▶ 477] | Get,Set | Attributwert. |

6.3.1.1.17 FB_AcsiBATypeINT128U

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeINT128U](#)

FUNCTION_BLOCK FB_AcsiBATypeINT128U EXTENDS FB_AcsiCommonAttributeClass

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|-------------------|---------|---------------|
| nValue | T_UINT128 [▶ 479] | Get,Set | Attributwert. |

6.3.1.1.18 FB_AcsiBATypeOCTETSTRING

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeOCTETSTRING](#)

FUNCTION_BLOCK FB_AcsiBATypeOCTETSTRING EXTENDS FB_AcsiCommonAttributeClass

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|---------|-------------|----------|--|
| bFixLen | BOOL | Get,Set | Attribut-Datenlänge (FALSE:=Variable Länge (Standardwert), TRUE:=Fixe Länge). |
| sAny | T_MaxString | Get, Set | Attributwert als formatierter Hexadezimalstring. Beispiele: „AA BB CC DD 01 02“ oder „00 12 34“. |

6.3.1.1.19 FB_AcsiBATypeUNICODESTRING

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeUNICODESTRING](#)

FUNCTION_BLOCK FB_AcsiBATypeUNICODESTRING EXTENDS FB_AcsiCommonAttributeClass

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|---------|--------------|----------|---|
| bFixLen | BOOL | Get,Set | Attribut-Datenlänge (FALSE:=Variable Länge (Standardwert), TRUE:=Fixe Länge). |
| sAny | WSTRING(255) | Get, Set | Attributwert. |

6.3.1.1.20 FB_AcsiBATypeVISIBLESTRING

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeVISIBLESTRING](#)

```
FUNCTION_BLOCK FB_AcsiBATypeVISIBLESTRING EXTENDS FB_AcsiCommonAttributeClass
```

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|---------|-------------|----------|---|
| bFixLen | BOOL | Get,Set | Attribut-Datenlänge (FALSE:=Variable Länge (Standardwert), TRUE:=Fixe Länge). |
| sAny | T_MaxString | Get, Set | Attributwert. |

6.3.1.1.21 FB_AcsiDATypeAddr

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBATypeOCTETSTRING](#) [[▶ 103](#)] -> [FB_AcsiDATypeAddr](#)

```
FUNCTION_BLOCK FB_AcsiDATypeAddr EXTENDS FB_AcsiBATypeOCTETSTRING
```

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| oValue | T_OCTET6 [▶ 478] | Get,Set | Attributwert. |

6.3.1.1.22 FB_AcsiDATypeAdjSt

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBATypeENUMERATED](#) [[▶ 98](#)] -> [FB_AcsiDATypeAdjSt](#)

```
FUNCTION_BLOCK FB_AcsiDATypeAdjSt EXTENDS FB_AcsiBATypeENUMERATED
```

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|---|---------|---------------|
| eValue | E_AcsiAdjSt [▶ 396] | Get,Set | Attributwert. |

6.3.1.1.23 FB_AcsiDATypeAlmLev

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBATypeENUMERATED](#) [[▶ 98](#)] -> [FB_AcsiDATypeAlmLev](#)

```
FUNCTION_BLOCK FB_AcsiDATypeAlmLev EXTENDS FB_AcsiBATypeENUMERATED
```

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiAlmLev [▶ 396] | Get,Set | Attributwert. |

6.3.1.1.24 FB_AcsiDATypeAnalogueValue

Namensraum: Tc3_Acsi [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

FB_AcsiCommonNodeClass [[▶ 154](#)] -> FB_AcsiCommonDataClass [[▶ 148](#)] -> FB_AcsiCommonAttributeClass [[▶ 146](#)] -> FB_AcsiDATypeAnalogueValue

FUNCTION_BLOCK FB_AcsiDATypeAnalogueValue EXTENDS FB_AcsiCommonAttributeClass

 **Eingänge**

| Name | Typ | Beschreibung |
|------|--|---|
| i | FB_AcsiBTypeINT32 [▶ 101] | Attribut zur Darstellung des Analogwertes als Ganzzahl. |
| f | FB_AcsiBTypeFLOAT32 [▶ 99] | Attribut zur Darstellung des Analogwertes als Fließkommazahl. |

 **Methoden**

| Name | Beschreibung |
|--|--|
| GetFloatPVal [▶ 105] | Liest den Fließkommazahl-Prozesswert. |
| GetIntPVal [▶ 106] | Liest den Ganzzahl-Prozesswert. |
| SetFloatPVal [▶ 106] | Schreibt den Fließkommazahl-Prozesswert. |
| SetIntPVal [▶ 106] | Schreibt den Ganzzahl-Prozesswert. |
| SetPVal [▶ 106] | Schreibt den Fließkommazahl- und Ganzzahl-Prozesswert. |

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|-------------------------------------|-------|---------|-----------------------------|
| bPresence [▶ 313] | BOOL | Get,Set | Attribut-Sichtbarkeit. |
| Fc [▶ 313] | DWORD | Get,Set | Funktionale Gruppe. |
| fValue | REAL | Get,Set | Wert des Datenattributs: f. |
| iValue | DINT | Get,Set | Wert des Datenattributs: i. |

6.3.1.1.24.1 GetFloatPVal

Liest das Attribut „f“ von „AnalogueValue“ (Fließkommazahldarstellung des Analogwertes) und skaliert den Attributwert in einen TwinCAT Prozesswert vom Typ: LREAL.

```
METHOD GetFloatPVal : BOOL
VAR_INPUT
    ipUnit : I_AcsiDATypeUnit;
END_VAR
VAR_OUTPUT
    fProcess : LREAL;
END_VAR
```

6.3.1.1.24.2 GetIntPVal

Liest das Attribut „i“ von „AnalogueValue“ (Ganzzahldarstellung des Analogwertes) und skaliert den Attributwert in einen TwinCAT Prozesswert vom Typ: LREAL.

```
METHOD GetIntPVal : BOOL
VAR_INPUT
    ipSVC      : I_AcsiDATypeScaledValueConfig;
END_VAR
VAR_OUTPUT
    fProcess   : LREAL;
END_VAR
```

6.3.1.1.24.3 SetFloatPVal

Skaliert einen TwinCAT Prozesswert vom Typ: LREAL und schreibt den Attributwert in das Attribut „f“ von „AnalogueValue“ (Fließkommazahldarstellung des Analogwertes).

```
METHOD SetFloatPVal : BOOL
VAR_INPUT
    fProcess   : LREAL;
    ipUnit     : I_AcsiDATypeUnit;
END_VAR
```

6.3.1.1.24.4 SetIntPVal

Skaliert einen TwinCAT Prozesswert vom Typ: LREAL und schreibt den Attributwert in das Attribut „i“ von „AnalogueValue“ (Ganzzahldarstellung des Analogwertes).

```
METHOD SetIntPVal : BOOL
VAR_INPUT
    fProcess   : LREAL;
    ipSVC      : I_AcsiDATypeScaledValueConfig;
END_VAR
```

6.3.1.1.24.5 SetPVal

Skaliert einen TwinCAT Prozesswert vom Typ: LREAL und schreibt den Attributwert in das Attribut „i“ und/oder „f“ von „AnalogueValue“ (Analogwert als Ganzzahl- oder Fließkommazahl).

```
METHOD SetPVal : BOOL
VAR_INPUT
    fProcess   : LREAL;
    ipUnit     : I_AcsiDATypeUnit;
    ipSVC      : I_AcsiDATypeScaledValueConfig;
END_VAR
```

6.3.1.1.25 FB_AcsiDATypeAngRef

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBATypeENUMERATED](#) [[▶ 98](#)] -> [FB_AcsiDATypeAngRef](#)

```
FUNCTION_BLOCK FB_AcsiDATypeAngRef EXTENDS FB_AcsiBATypeENUMERATED
```



Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiAngRef [▶ 397] | Get,Set | Attributwert. |

6.3.1.1.26 **FB_AcsiDATypeApplAddCause**

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDATypeApplAddCause](#)

FUNCTION_BLOCK FB_AcsiDATypeApplAddCause EXTENDS FB_AcsiBATypeENUMERATED



Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiApplAddCause [▶ 398] | Get,Set | Attributwert. |

6.3.1.1.27 **FB_AcsiDATypeApplError**

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDATypeApplError](#)

FUNCTION_BLOCK FB_AcsiDATypeApplError EXTENDS FB_AcsiBATypeENUMERATED



Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|---|---------|---------------|
| eValue | E_AcsiApplError [▶ 399] | Get,Set | Attributwert. |

6.3.1.1.28 **FB_AcsiDATypeAutoRecSt**

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDATypeAutoRecSt](#)

FUNCTION_BLOCK FB_AcsiDATypeAutoRecSt EXTENDS FB_AcsiBATypeENUMERATED



Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|---|---------|---------------|
| eValue | E_AcsiAutoRecSt [▶ 400] | Get,Set | Attributwert. |

6.3.1.1.29 **FB_AcsiDATypeBeh**

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDTypeBeh](#)

FUNCTION_BLOCK FB_AcsiDTypeBeh EXTENDS FB_AcsiBTypeENUMERATED

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|-----------------------------------|---------|---------------|
| eValue | E_AcsiBeh [▶ 400] | Get,Set | Attributwert. |

6.3.1.1.30 FB_AcsiDTypeCalendarTime

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiDTypeCalendarTime](#)

FUNCTION_BLOCK FB_AcsiDTypeCalendarTime EXTENDS FB_AcsiCommonAttributeClass

Eingänge

| Name | Typ | Beschreibung |
|---------|---|----------------|
| occ | FB_AcsiBTypeINT16U [▶ 100] | Datenattribut. |
| occType | FB_AcsiDTypeOccType [▶ 121] | Datenattribut. |
| occPer | FB_AcsiDTypeOccPer [▶ 121] | Datenattribut. |
| weekDay | FB_AcsiDTypeWeekDay [▶ 145] | Datenattribut. |
| month | FB_AcsiDTypeMonth [▶ 120] | Datenattribut. |
| day | FB_AcsiBTypeINT8U [▶ 100] | Datenattribut. |
| hr | FB_AcsiBTypeINT8U [▶ 100] | Datenattribut. |
| mn | FB_AcsiBTypeINT8U [▶ 100] | Datenattribut. |

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|-----------------------------------|---------------------------------------|---------|--------------------------|
| bPresence [▶ 313] | BOOL | Get,Set | Attribut-Sichtbarkeit. |
| Fc [▶ 313] | DWORD | Get,Set | Funktionale Gruppe. |
| nOcc | WORD | Get,Set | Wert des Datenattributs. |
| eOccType | E_AcsiOccType [▶ 417] | Get,Set | Wert des Datenattributs. |
| eOccPer | E_AcsiOccPer [▶ 417] | Get,Set | Wert des Datenattributs. |
| eWeekDay | E_AcsiWeekDay [▶ 442] | Get,Set | Wert des Datenattributs. |
| eMonth | E_AcsiMonth [▶ 415] | Get,Set | Wert des Datenattributs. |
| nDay | BYTE | Get,Set | Wert des Datenattributs. |
| nHr | BYTE | Get,Set | Wert des Datenattributs. |
| nMn | BYTE | Get,Set | Wert des Datenattributs. |

6.3.1.1.31 FB_AcsiDATypeCBOpCap

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDATypeCBOpCap](#)

FUNCTION_BLOCK FB_AcsiDATypeCBOpCap EXTENDS FB_AcsiBATypeENUMERATED

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|---------------------------------------|---------|---------------|
| eValue | E_AcsiCBOpCap [▶ 401] | Get,Set | Attributwert. |

6.3.1.1.32 FB_AcsiDATypeCell

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiDATypeCell](#)

FUNCTION_BLOCK FB_AcsiDATypeCell EXTENDS FB_AcsiCommonAttributeClass

Eingänge

| Name | Typ | Beschreibung |
|--------|--|----------------|
| xStart | FB_AcsiDATypeAnalogueValue [▶ 105] | Datenattribut. |
| xEnd | FB_AcsiDATypeAnalogueValue [▶ 105] | Datenattribut. |
| yStart | FB_AcsiDATypeAnalogueValue [▶ 105] | Datenattribut. |
| yEnd | FB_AcsiDATypeAnalogueValue [▶ 105] | Datenattribut. |

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|-----------------------------------|-------|---------|--------------------------|
| bPresence [▶ 313] | BOOL | Get,Set | Attribut-Sichtbarkeit. |
| Fc [▶ 313] | DWORD | Get,Set | Funktionale Gruppe. |
| fXStart | REAL | Get,Set | Wert des Datenattributs. |
| fXEnd | REAL | Get,Set | Wert des Datenattributs. |
| fYStart | REAL | Get,Set | Wert des Datenattributs. |
| fYEnd | REAL | Get,Set | Wert des Datenattributs. |

6.3.1.1.33 FB_AcsiDATypeCheck

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeCODEDENUM \[▶ 98\]](#) -> [FB_AcsiDTypeCheck](#)

FUNCTION_BLOCK FB_AcsiDTypeCheck EXTENDS FB_AcsiBTypeCODEDENUM

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|-------------------------------------|---------|--|
| eValue | E_AcsiCheck [▶ 401] | Get,Set | Attributwert als Aufzählungstyp. |
| b0 | BOOL | Get,Set | Attributwert. |
| b1 | BOOL | Get,Set | Attributwert. |
| nValue | BYTE(0..3) | Get,Set | Attributwert im TwinCAT SPS BYTE-Format. |

6.3.1.1.34 FB_AcsiDTypeClcIntvTyp

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDTypeClcIntvTyp](#)

FUNCTION_BLOCK FB_AcsiDTypeClcIntvTyp EXTENDS FB_AcsiBTypeENUMERATED

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiClcIntvTyp [▶ 402] | Get,Set | Attributwert. |

6.3.1.1.35 FB_AcsiDTypeClcMod

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDTypeClcMod](#)

FUNCTION_BLOCK FB_AcsiDTypeClcMod EXTENDS FB_AcsiBTypeENUMERATED

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|--------------------------------------|---------|---------------|
| eValue | E_AcsiClcMod [▶ 402] | Get,Set | Attributwert. |

6.3.1.1.36 FB_AcsiDTypeClcMth

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDTypeClcMth](#)

FUNCTION_BLOCK FB_AcsiDTypeClcMth EXTENDS FB_AcsiBTypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiClcMth [▶ 403] | Get,Set | Attributwert. |

6.3.1.1.37 FB_AcsiDATypeClcRfTyp

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBTypeENUMERATED](#) [[▶ 98](#)] -> [FB_AcsiDATypeClcRfTyp](#)

FUNCTION_BLOCK FB_AcsiDATypeClcRfTyp EXTENDS FB_AcsiBTypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiClcRfTyp [▶ 403] | Get,Set | Attributwert. |

6.3.1.1.38 FB_AcsiDATypeClcTotVA

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBTypeENUMERATED](#) [[▶ 98](#)] -> [FB_AcsiDATypeClcTotVA](#)

FUNCTION_BLOCK FB_AcsiDATypeClcTotVA EXTENDS FB_AcsiBTypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiClcTotVA [▶ 404] | Get,Set | Attributwert. |

6.3.1.1.39 FB_AcsiDATypeCmdQual

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBTypeENUMERATED](#) [[▶ 98](#)] -> [FB_AcsiDATypeCmdQual](#)

FUNCTION_BLOCK FB_AcsiDATypeCmdQual EXTENDS FB_AcsiBTypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|---|---------|---------------|
| eValue | E_AcsiCmdQual [▶ 404] | Get,Set | Attributwert. |

6.3.1.1.40 FB_AcsiDATypeCtlModels

Namensraum: [Tc3_Acsi](#) [[96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[154](#)] -> [FB_AcsiCommonDataClass](#) [[148](#)] -> [FB_AcsiCommonAttributeClass](#) [[146](#)] -> [FB_AcsiBATypeENUMERATED](#) [[98](#)] -> [FB_AcsiDATypeCtlModels](#)

```
FUNCTION_BLOCK FB_AcsiDATypeCtlModels EXTENDS FB_AcsiBATypeENUMERATED
```

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|---|---------|---------------|
| eValue | E_AcsiCtlModels [405] | Get,Set | Attributwert. |

6.3.1.1.41 FB_AcsiDATypeCurrency

Namensraum: [Tc3_Acsi](#) [[96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[154](#)] -> [FB_AcsiCommonDataClass](#) [[148](#)] -> [FB_AcsiCommonAttributeClass](#) [[146](#)] -> [FB_AcsiBATypeVISIBLESTRING](#) [[103](#)] -> [FB_AcsiDATypeCurrency](#)

```
FUNCTION_BLOCK FB_AcsiDATypeCurrency EXTENDS FB_AcsiBATypeVISIBLESTRING
```

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|-----------|---------|---------------|
| sValue | STRING(3) | Get,Set | Attributwert. |

6.3.1.1.42 FB_AcsiDATypeCycTrMod

Namensraum: [Tc3_Acsi](#) [[96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[154](#)] -> [FB_AcsiCommonDataClass](#) [[148](#)] -> [FB_AcsiCommonAttributeClass](#) [[146](#)] -> [FB_AcsiBATypeENUMERATED](#) [[98](#)] -> [FB_AcsiDATypeCycTrMod](#)

```
FUNCTION_BLOCK FB_AcsiDATypeCycTrMod EXTENDS FB_AcsiBATypeENUMERATED
```

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiCycTrMod [406] | Get,Set | Attributwert. |

6.3.1.1.43 FB_AcsiDATypeDbpos

Namensraum: [Tc3_Acsi](#) [[96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeCODEDENUM \[▶ 98\]](#) -> [FB_AcsiDTypeDbpos](#)

FUNCTION_BLOCK FB_AcsiDTypeDbpos EXTENDS FB_AcsiBTypeCODEDENUM

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|-------------------------------------|---------|--|
| eValue | E_AcsiDbpos [▶ 406] | Get,Set | Attributwert als Aufzählungstyp. |
| b0 | BOOL | Get,Set | Attributwert. |
| b1 | BOOL | Get,Set | Attributwert. |
| nValue | BYTE(0..3) | Get,Set | Attributwert im TwinCAT SPS BYTE-Format. |

6.3.1.1.44 FB_AcsiDTypeDir

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDTypeDir](#)

FUNCTION_BLOCK FB_AcsiDTypeDir EXTENDS FB_AcsiBTypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|-----------------------------------|---------|---------------|
| eValue | E_AcsiDir [▶ 407] | Get,Set | Attributwert. |

6.3.1.1.45 FB_AcsiDTypeDirMod

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDTypeDirMod](#)

FUNCTION_BLOCK FB_AcsiDTypeDirMod EXTENDS FB_AcsiBTypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|--------------------------------------|---------|---------------|
| eValue | E_AcsiDirMod [▶ 407] | Get,Set | Attributwert. |

6.3.1.1.46 FB_AcsiDTypeEEHealth

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDTypeEEHealth](#)

FUNCTION_BLOCK FB_AcsiDTypeEEHealth EXTENDS FB_AcsiBTypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiEEHealth [▶ 408] | Get,Set | Attributwert. |

6.3.1.1.47 **FB_AcsiDATypeEntryID**

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBATypeOCTETSTRING](#) [[▶ 103](#)] -> **FB_AcsiDATypeEntryID**

FUNCTION_BLOCK FB_AcsiDATypeEntryID EXTENDS FB_AcsiBATypeOCTETSTRING

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---|
| nValue | LWORD | Get,Set | <p>Attributwert im TwinCAT SPS LWORD-Format.</p> <p>Die 8 Bytes des OctetStrings können als Hexadezimalwert gesetzt oder abgefragt werden.</p> <p>Beispiele:</p> <p>LWORD#16#1122334455667788 entspricht OctetString: „1 1 22 33 44 55 66 77 88“.</p> <p>LWORD#16#000000000000AABB entspricht OctetString: „00 00 00 00 00 00 AA BB“.</p> <p>LWORD#16#AABB000000000000 entspricht OctetString: „AA BB 00 00 00 00 00 00“.</p> |
| oValue | T_OCTET8 [▶ 478] | Get,Set | Attributwert. |

6.3.1.1.48 **FB_AcsiDATypeEntryTime**

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> **FB_AcsiDATypeEntryTime**

FUNCTION_BLOCK FB_AcsiDATypeEntryTime EXTENDS FB_AcsiCommonAttributeClass

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|-----------|---|---------|--|
| tValue | T_BinaryTime
[▶ 476] | Get,Set | Attributwert als strukturierter Typ. |
| TimeOfDay | TOD | Get,Set | Attributwert: Anzahl der vergangenen Millisekunden seit Mitternacht des aktuellen Tages im TwinCAT SPS TOD-Format. |
| Day | WORD | Get,Set | Attributwert: Anzahl der vergangenen Tage seit dem 1. Januar 1984 im TwinCAT SPS WORD-Format. |
| D | DATE | Get,Set | Attributwert: Anzahl der vergangenen Tage seit dem 1. Januar 1984 im TwinCAT SPS DATE-Format. |
| sValue | STRING(26) | Get,Set | Attributwert: Binary-Time Zeitinformation als String mit folgendem Format: „BT#1984-01-01-00:00:00.000“. |
| nValue | LWORD(0..16#FFF
FFFFFFFF) | Get,Set | Attributwert im TwinCAT SPS LWORD-Format. (nur die untersten 48 Bits werden verwendet). |

6.3.1.1.49 FB_AcsiDATypeFailMod

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeENUMERATED \[▶ 98\]](#) -> **FB_AcsiDATypeFailMod**

FUNCTION_BLOCK FB_AcsiDATypeFailMod EXTENDS FB_AcsiBTypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|---------------------------------------|---------|---------------|
| eValue | E_AcsiFailMod [▶ 408] | Get,Set | Attributwert. |

6.3.1.1.50 FB_AcsiDATypeFanCtl

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeENUMERATED \[▶ 98\]](#) -> **FB_AcsiDATypeFanCtl**

FUNCTION_BLOCK FB_AcsiDATypeFanCtl EXTENDS FB_AcsiBTypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|--------------------------------------|---------|---------------|
| eValue | E_AcsiFanCtl [▶ 408] | Get,Set | Attributwert. |

6.3.1.1.51 FB_AcsiDATypeFanCtlGen

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDTypeFanCtlGen](#)

FUNCTION_BLOCK FB_AcsiDTypeFanCtlGen EXTENDS FB_AcsiBTypeENUMERATED

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|---|---------|---------------|
| eValue | E_AcsiFanCtlGen [▶ 409] | Get,Set | Attributwert. |

6.3.1.1.52 **FB_AcsiDTypeFilTyp**

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDTypeFilTyp](#)

FUNCTION_BLOCK FB_AcsiDTypeFilTyp EXTENDS FB_AcsiBTypeENUMERATED

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|--------------------------------------|---------|---------------|
| eValue | E_AcsiFilTyp [▶ 410] | Get,Set | Attributwert. |

6.3.1.1.53 **FB_AcsiDTypeFltLoop**

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDTypeFltLoop](#)

FUNCTION_BLOCK FB_AcsiDTypeFltLoop EXTENDS FB_AcsiBTypeENUMERATED

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|---------------------------------------|---------|---------------|
| eValue | E_AcsiFltLoop [▶ 411] | Get,Set | Attributwert. |

6.3.1.1.54 **FB_AcsiDTypeGnSt**

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDTypeGnSt](#)

FUNCTION_BLOCK FB_AcsiDTypeGnSt EXTENDS FB_AcsiBTypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiGnSt [▶ 411] | Get,Set | Attributwert. |

6.3.1.1.55 FB_AcsiDATypeHealth

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBTypeENUMERATED](#) [[▶ 98](#)] -> [FB_AcsiDATypeHealth](#)

FUNCTION_BLOCK FB_AcsiDATypeHealth EXTENDS FB_AcsiBTypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiHealth [▶ 412] | Get,Set | Attributwert. |

6.3.1.1.56 FB_AcsiDATypeHvRef

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBTypeENUMERATED](#) [[▶ 98](#)] -> [FB_AcsiDATypeHvRef](#)

FUNCTION_BLOCK FB_AcsiDATypeHvRef EXTENDS FB_AcsiBTypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|---|---------|---------------|
| eValue | E_AcsiHvRef [▶ 412] | Get,Set | Attributwert. |

6.3.1.1.57 FB_AcsiDATypeIntrDetMth

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBTypeENUMERATED](#) [[▶ 98](#)] -> [FB_AcsiDATypeIntrDetMth](#)

FUNCTION_BLOCK FB_AcsiDATypeIntrDetMth EXTENDS FB_AcsiBTypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiIntrDetMth [▶ 412] | Get,Set | Attributwert. |

6.3.1.1.58 FB_AcsiDATypeLastAppLError

Namensraum: Tc3_Acsi [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

FB_AcsiCommonNodeClass [[▶ 154](#)] -> FB_AcsiCommonDataClass [[▶ 148](#)] -> FB_AcsiCommonAttributeClass [[▶ 146](#)] -> FB_AcsiDATypeAppLError

```
FUNCTION_BLOCK FB_AcsiDATypeLastAppLError EXTENDS FB_AcsiCommonAttributeClass IMPLEMENTS I_AcsiDATypeLastAppLError
```

Schnittstellen

| Typ | Beschreibung |
|---|---|
| I_AcsiDATypeLastAppLError [▶ 322] | Schnittstellenzeiger der LastAppLError-Objektinstanz. |

Eingänge

| Name | Typ | Beschreibung |
|----------|---|----------------|
| CntrlObj | FB_AcsiDATypeVisString129 [▶ 144] | Datenattribut. |
| Error | FB_AcsiDATypeAppLError [▶ 107] | Datenattribut. |
| Origin | FB_AcsiDATypeOriginator [▶ 125] | Datenattribut. |
| ctlNum | FB_AcsiBTypeINT8U [▶ 100] | Datenattribut. |
| AddCause | FB_AcsiDATypeApplAddCause [▶ 107] | Datenattribut. |

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|-------------------------------------|--|---------|------------------------|
| bPresence [▶ 313] | BOOL | Get,Set | Attribut-Sichtbarkeit. |
| Fc [▶ 313] | DWORD | Get,Set | Funktionale Gruppe. |
| sCntrlObj | STRING(129) | Get,Set | Attributwert. |
| eError | E_AcsiAppLError [▶ 399] | Get,Set | Attributwert. |
| stOrigin | ST_AcsiOriginator [▶ 460] | Get,Set | Attributwert. |
| nCtlNum | BYTE | Get,Set | Attributwert. |
| eAddCause | E_AcsiApplAddCause [▶ 398] | Get,Set | Attributwert. |

6.3.1.1.59 FB_AcsiDATypeLevMod

Namensraum: Tc3_Acsi [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

FB_AcsiCommonNodeClass [[▶ 154](#)] -> FB_AcsiCommonDataClass [[▶ 148](#)] -> FB_AcsiCommonAttributeClass [[▶ 146](#)] -> FB_AcsiBTypeENUMERATED [[▶ 98](#)] -> FB_AcsiDATypeLevMod

```
FUNCTION_BLOCK FB_AcsiDATypeLevMod EXTENDS FB_AcsiBTypeENUMERATED
```

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiLevMod [▶ 413] | Get,Set | Attributwert. |

6.3.1.1.60 FB_AcsiDATypeLivDeaMod

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBTypeENUMERATED](#) [[▶ 98](#)] -> [FB_AcsiDATypeLivDeaMod](#)

FUNCTION_BLOCK FB_AcsiDATypeLivDeaMod EXTENDS FB_AcsiBTypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|---|---------|---------------|
| eValue | E_AcsiLivDeaMod [▶ 413] | Get,Set | Attributwert. |

6.3.1.1.61 FB_AcsiDATypeMechHealth

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBTypeENUMERATED](#) [[▶ 98](#)] -> [FB_AcsiDATypeMechHealth](#)

FUNCTION_BLOCK FB_AcsiDATypeMechHealth EXTENDS FB_AcsiBTypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiMechHealth [▶ 415] | Get,Set | Attributwert. |

6.3.1.1.62 FB_AcsiDATypeMod

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)]-> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBTypeENUMERATED](#) [[▶ 98](#)] -> [FB_AcsiDATypeMod](#)

FUNCTION_BLOCK FB_AcsiDATypeMod EXTENDS FB_AcsiBTypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|---|---------|---------------|
| eValue | E_AcsiMod [▶ 415] | Get,Set | Attributwert. |

6.3.1.1.63 FB_AcsiDATypeMonth

Namensraum: [Tc3_Acsi](#) [[96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[154](#)] -> [FB_AcsiCommonDataClass](#) [[148](#)] -> [FB_AcsiCommonAttributeClass](#) [[146](#)] -> [FB_AcsiBATypeENUMERATED](#) [[98](#)] -> [FB_AcsiDATypeMonth](#)

```
FUNCTION_BLOCK FB_AcsiDATypeMonth EXTENDS FB_AcsiBATypeENUMERATED
```

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|---|---------|---------------|
| eValue | E_AcsiMonth [415] | Get,Set | Attributwert. |

6.3.1.1.64 FB_AcsiDATypeMultiplier

Namensraum: [Tc3_Acsi](#) [[96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[154](#)] -> [FB_AcsiCommonDataClass](#) [[148](#)] -> [FB_AcsiCommonAttributeClass](#) [[146](#)] -> [FB_AcsiBATypeENUMERATED](#) [[98](#)] -> [FB_AcsiDATypeMultiplier](#)

```
FUNCTION_BLOCK FB_AcsiDATypeMultiplier EXTENDS FB_AcsiBATypeENUMERATED
```

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiMultiplier [416] | Get,Set | Attributwert. |

6.3.1.1.65 FB_AcsiDATypeObjectName

Namensraum: [Tc3_Acsi](#) [[96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[154](#)] -> [FB_AcsiCommonDataClass](#) [[148](#)] -> [FB_AcsiCommonAttributeClass](#) [[146](#)] -> [FB_AcsiBATypeVISIBLESTRING](#) [[103](#)] -> [FB_AcsiDATypeObjectName](#)

```
FUNCTION_BLOCK FB_AcsiDATypeObjectName EXTENDS FB_AcsiBATypeVISIBLESTRING
```

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| sValue | T_AcsiObjectName [475] | Get,Set | Attributwert. |

6.3.1.1.66 FB_AcsiDATypeObjRef

Namensraum: [Tc3_Acsi](#) [[96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeVISIBLESTRING \[▶ 103\]](#) -> [FB_AcsiDTypeObjRef](#)

FUNCTION_BLOCK FB_AcsiDTypeObjRef EXTENDS FB_AcsiBTypeVISIBLESTRING

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|---|---------|---------------|
| sValue | T_AcsiObjectReference [▶ 476] | Get,Set | Attributwert. |

6.3.1.1.67 FB_AcsiDTypeOccPer

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDTypeOccPer](#)

FUNCTION_BLOCK FB_AcsiDTypeOccPer EXTENDS FB_AcsiBTypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|--------------------------------------|---------|---------------|
| eValue | E_AcsiOccPer [▶ 417] | Get,Set | Attributwert. |

6.3.1.1.68 FB_AcsiDTypeOccType

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDTypeOccType](#)

FUNCTION_BLOCK FB_AcsiDTypeOccType EXTENDS FB_AcsiBTypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|---------------------------------------|---------|---------------|
| eValue | E_AcsiOccType [▶ 417] | Get,Set | Attributwert. |

6.3.1.1.69 FB_AcsiDTypeOctet16

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeOCTETSTRING \[▶ 103\]](#) -> [FB_AcsiDTypeOctet16](#)

FUNCTION_BLOCK FB_AcsiDTypeOctet16 EXTENDS FB_AcsiBTypeOCTETSTRING

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|---|---------|---------------|
| oValue | T_OCTET16 [▶ 478] | Get,Set | Attributwert. |

6.3.1.1.70 FB_AcsiDATypeOctet32

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBTypeOCTETSTRING](#) [[▶ 103](#)] -> [FB_AcsiDATypeOctet32](#)

FUNCTION_BLOCK FB_AcsiDATypeOctet32 EXTENDS FB_AcsiBTypeOCTETSTRING

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|---|---------|---------------|
| oValue | T_OCTET32 [▶ 479] | Get,Set | Attributwert. |

6.3.1.1.71 FB_AcsiDATypeOctet64

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBTypeOCTETSTRING](#) [[▶ 103](#)] -> [FB_AcsiDATypeOctet64](#)

FUNCTION_BLOCK FB_AcsiDATypeOctet64 EXTENDS FB_AcsiBTypeOCTETSTRING

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|---|---------|---------------|
| oValue | T_OCTET64 [▶ 479] | Get,Set | Attributwert. |

6.3.1.1.72 FB_AcsiDATypeOctet8

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBTypeOCTETSTRING](#) [[▶ 103](#)] -> [FB_AcsiDATypeOctet8](#)

FUNCTION_BLOCK FB_AcsiDATypeOctet8 EXTENDS FB_AcsiBTypeOCTETSTRING

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| oValue | T_OCTET8 [▶ 478] | Get,Set | Attributwert. |

6.3.1.1.73 **FB_AcsiDATypeOpModRect**

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeENUMERATED \[▶ 98\]](#) -> **FB_AcsiDATypeOpModRect**

FUNCTION_BLOCK FB_AcsiDATypeOpModRect EXTENDS FB_AcsiBATypeENUMERATED



Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|---|---------|---------------|
| eValue | E_AcsiOpModRect [▶ 418] | Get,Set | Attributwert. |

6.3.1.1.74 **FB_AcsiDATypeOpModSyn**

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeENUMERATED \[▶ 98\]](#) -> **FB_AcsiDATypeOpModSyn**

FUNCTION_BLOCK FB_AcsiDATypeOpModSyn EXTENDS FB_AcsiBATypeENUMERATED



Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiOpModSyn [▶ 418] | Get,Set | Attributwert. |

6.3.1.1.75 **FB_AcsiDATypeOptFlds**

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> **FB_AcsiDATypeOptFlds**

FUNCTION_BLOCK FB_AcsiDATypeOptFlds EXTENDS FB_AcsiCommonAttributeClass

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------------------|----------------------------------|---------|---|
| cValue | ST_AcsiOptionalFields
[▶ 459] | Get,Set | Attributwert als strukturierter Typ. |
| SequenceNumber | BOOL | Get,Set | Attributwert. |
| ReportTimeStamp | BOOL | Get,Set | Attributwert. |
| ReasonForInclusion | BOOL | Get,Set | Attributwert. |
| DataSetName | BOOL | Get,Set | Attributwert. |
| DataReference | BOOL | Get,Set | Attributwert. |
| BufferOverflow | BOOL | Get,Set | Attributwert. |
| EntryID | BOOL | Get,Set | Attributwert. |
| ConfRevision | BOOL | Get,Set | Attributwert. |
| Segmentation | BOOL | Get,Set | Attributwert. |
| nValue | WORD(0..1023) | Get,Set | Information über die optionalen Felder die im Report übertragen werden im TwinCAT SPS WORD-Format (nur die unteren 10 Bits werden verwendet). |

6.3.1.1.76 FB_AcsiDATypeOrCategory

Namensraum: Tc3_Acsi [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

FB_AcsiCommonNodeClass [▶ 154] -> FB_AcsiCommonDataClass [▶ 148] -> FB_AcsiCommonAttributeClass [▶ 146] -> FB_AcsiBTypeENUMERATED [▶ 98] -> FB_AcsiDATypeOrCategory

```
FUNCTION_BLOCK FB_AcsiDATypeOrCategory EXTENDS FB_AcsiBTypeENUMERATED
```

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|-----------------------------|---------|---------------|
| eValue | E_AcsiOrCategory
[▶ 419] | Get,Set | Attributwert. |

6.3.1.1.77 FB_AcsiDATypeOrIdent

Namensraum: Tc3_Acsi [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

FB_AcsiCommonNodeClass [▶ 154] -> FB_AcsiCommonDataClass [▶ 148] -> FB_AcsiCommonAttributeClass [▶ 146] -> FB_AcsiBTypeOCTETSTRING [▶ 103] -> FB_AcsiDATypeOrIdent

```
FUNCTION_BLOCK FB_AcsiDATypeOrIdent EXTENDS FB_AcsiBTypeOCTETSTRING
```

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|-------------------|---------|---------------|
| oValue | T_OCTET64 [▶ 479] | Get,Set | Attributwert. |

6.3.1.1.78 FB_AcsiDATypeOriginator

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiDATypeOriginator](#)

FUNCTION_BLOCK FB_AcsiDATypeOriginator EXTENDS FB_AcsiCommonAttributeClass

Eingänge

| Name | Typ | Beschreibung |
|---------|---|----------------|
| orCat | FB_AcsiDATypeOrCategory [▶ 124] | Datenattribut. |
| orIdent | FB_AcsiDATypeOrIdent [▶ 124] | Datenattribut. |

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|-----------------------------------|---|---------|--------------------------------------|
| bPresence [▶ 313] | BOOL | Get,Set | Attribut-Sichtbarkeit. |
| Fc [▶ 313] | DWORD | Get,Set | Funktionale Gruppe. |
| stOrigin | ST_AcsiOriginator [▶ 460] | Get,Set | Attributwert als strukturierter Typ. |
| eOrCat | E_AcsiOrCategory [▶ 419] | Get,Set | Attributwert. |
| eOrIdent | T_OCTET64 [▶ 479] | Get,Set | Attributwert. |

6.3.1.1.79 FB_AcsiDATypeParColMod

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDATypeParColMod](#)

FUNCTION_BLOCK FB_AcsiDATypeParColMod EXTENDS FB_AcsiBATypeENUMERATED

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|---|---------|---------------|
| eValue | E_AcsiParColMod [▶ 419] | Get,Set | Attributwert. |

6.3.1.1.80 FB_AcsiDATypeParMod

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDTypeParMod](#)

FUNCTION_BLOCK FB_AcsiDTypeParMod EXTENDS FB_AcsiBTypeENUMERATED

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|--------------------------------------|---------|---------------|
| eValue | E_AcsiParMod [▶ 420] | Get,Set | Attributwert. |

6.3.1.1.81 **FB_AcsiDTypeParTraMod**

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDTypeParTraMod](#)

FUNCTION_BLOCK FB_AcsiDTypeParTraMod EXTENDS FB_AcsiBTypeENUMERATED

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|---|---------|---------------|
| eValue | E_AcsiParTraMod [▶ 420] | Get,Set | Attributwert. |

6.3.1.1.82 **FB_AcsiDTypePFSign**

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDTypePFSign](#)

FUNCTION_BLOCK FB_AcsiDTypePFSign EXTENDS FB_AcsiBTypeENUMERATED

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|--------------------------------------|---------|---------------|
| eValue | E_AcsiPFSign [▶ 421] | Get,Set | Attributwert. |

6.3.1.1.83 **FB_AcsiDTypePhsRef**

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDTypePhsRef](#)

FUNCTION_BLOCK FB_AcsiDTypePhsRef EXTENDS FB_AcsiBTypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiPhsRef [▶ 421] | Get,Set | Attributwert. |

6.3.1.1.84 FB_AcsiDATypePhyComAddr

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiDATypePhyComAddr](#)

FUNCTION_BLOCK FB_AcsiDATypePhyComAddr EXTENDS FB_AcsiCommonAttributeClass

 **Eingänge**

| Name | Typ | Beschreibung |
|----------|--|----------------|
| Addr | FB_AcsiDATypeAddr [▶ 104] | Datenattribut. |
| PRIORITY | FB_AcsiBTypeINT8U [▶ 100] | Datenattribut. |
| VID | FB_AcsiBTypeINT16U [▶ 100] | Datenattribut. |
| APPID | FB_AcsiBTypeINT16U [▶ 100] | Datenattribut. |

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|---|------------------------------------|---------|------------------------|
| bPresence [▶ 313] | BOOL | Get,Set | Attribut-Sichtbarkeit. |
| Fc [▶ 313] | DWORD | Get,Set | Funktionale Gruppe. |
| oAddr | T_OCTET6 [▶ 478] | Get,Set | Attributwert. |
| nPRIORITY | BYTE(0..7) | Get,Set | Attributwert. |
| nVID | WORD(0..4095) | Get,Set | Attributwert. |
| nAPPID | WORD | Get,Set | Attributwert. |

6.3.1.1.85 FB_AcsiDATypePhyHealth

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBTypeENUMERATED](#) [[▶ 98](#)] -> [FB_AcsiDATypePhyHealth](#)

FUNCTION_BLOCK FB_AcsiDATypePhyHealth EXTENDS FB_AcsiBTypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|---|---------|---------------|
| eValue | E_AcsiPhyHealth [▶ 422] | Get,Set | Attributwert. |

6.3.1.1.86 FB_AcsiDATypePIDAlg

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDATypePIDAlg](#)

FUNCTION_BLOCK FB_AcsiDATypePIDAlg EXTENDS FB_AcsiBATypeENUMERATED

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|--------------------------------------|---------|---------------|
| eValue | E_AcsiPIDAlg [▶ 422] | Get,Set | Attributwert. |

6.3.1.1.87 FB_AcsiDATypePmpCtl

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDATypePmpCtl](#)

FUNCTION_BLOCK FB_AcsiDATypePmpCtl EXTENDS FB_AcsiBATypeENUMERATED

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|--------------------------------------|---------|---------------|
| eValue | E_AcsiPmpCtl [▶ 422] | Get,Set | Attributwert. |

6.3.1.1.88 FB_AcsiDATypePmpCtlGen

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDATypePmpCtlGen](#)

FUNCTION_BLOCK FB_AcsiDATypePmpCtlGen EXTENDS FB_AcsiBATypeENUMERATED

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|---|---------|---------------|
| eValue | E_AcsiPmpCtlGen [▶ 423] | Get,Set | Attributwert. |

6.3.1.1.89 FB_AcsiDATypePoint

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiDATypePoint](#)

FUNCTION_BLOCK FB_AcsiDATypePoint EXTENDS FB_AcsiCommonAttributeClass

 **Eingänge**

| Name | Typ | Beschreibung |
|------|--|----------------|
| xVal | FB_AcsiDATypeAnalogueValue [▶ 105] | Datenattribut. |
| yVal | FB_AcsiDATypeAnalogueValue [▶ 105] | Datenattribut. |
| zVal | FB_AcsiDATypeAnalogueValue [▶ 105] | Datenattribut. |

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------------------------------------|-------|---------|------------------------|
| bPresence
[▶ 313] | BOOL | Get,Set | Attribut-Sichtbarkeit. |
| Fc [▶ 313] | DWORD | Get,Set | Funktionale Gruppe. |
| fXVal | REAL | Get,Set | Attributwert. |
| fYVal | REAL | Get,Set | Attributwert. |
| fZVal | REAL | Get,Set | Attributwert. |

6.3.1.1.90 FB_AcsiDATypePolQty

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDATypePolQty](#)

FUNCTION_BLOCK FB_AcsiDATypePolQty EXTENDS FB_AcsiBATypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|--------------------------------------|---------|---------------|
| eValue | E_AcsiPolQty [▶ 423] | Get,Set | Attributwert. |

6.3.1.1.91 FB_AcsiDATypePOWCap

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDATypePOWCap](#)

FUNCTION_BLOCK FB_AcsiDATypePOWCap EXTENDS FB_AcsiBATypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|--------------------------------------|---------|---------------|
| eValue | E_AcsiPOWCap [▶ 424] | Get,Set | Attributwert. |

6.3.1.1.92 FB_AcsiDATypePulseConfig

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiDATypePulseConfig](#)

```
FUNCTION_BLOCK FB_AcsiDATypePulseConfig EXTENDS FB_AcsiCommonAttributeClass
```

Eingänge

| Name | Typ | Beschreibung |
|---------|--|----------------|
| cmdQual | FB_AcsiDATypeCmdQual [▶ 111] | Datenattribut. |
| onDur | FB_AcsiBTypeINT32U [▶ 101] | Datenattribut. |
| offDur | FB_AcsiBTypeINT32U [▶ 101] | Datenattribut. |
| numPIs | FB_AcsiBTypeINT32U [▶ 101] | Datenattribut. |

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|-----------------------------------|---------------------------------------|---------|------------------------|
| bPresence [▶ 313] | BOOL | Get,Set | Attribut-Sichtbarkeit. |
| Fc [▶ 313] | DWORD | Get,Set | Funktionale Gruppe. |
| eCmdQual | E_AcsiCmdQual [▶ 404] | Get,Set | Attributwert. |
| nOnDur | DWORD | Get,Set | Attributwert. |
| nOffDur | DWORD | Get,Set | Attributwert. |
| nNumPIs | DWORD | Get,Set | Attributwert. |

6.3.1.1.93 FB_AcsiDATypeQuality

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiDATypeQuality](#)

```
FUNCTION_BLOCK FB_AcsiDATypeQuality EXTENDS FB_AcsiCommonAttributeClass
```

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|-----------------|---|---------|--|
| qValue | ST_AcsiQuality [▶ 463] | Get,Set | Attributwert als strukturierter Typ. |
| eValidity | E_AcsiQualityValidity [▶ 425] | Get,Set | Attributwert. |
| Overflow | BOOL | Get,Set | Attributwert. |
| OutOfRange | BOOL | Get,Set | Attributwert. |
| BadReference | BOOL | Get,Set | Attributwert. |
| Oscillatory | BOOL | Get,Set | Attributwert. |
| Failure | BOOL | Get,Set | Attributwert. |
| OldData | BOOL | Get,Set | Attributwert. |
| Inconsistent | BOOL | Get,Set | Attributwert. |
| Inaccurate | BOOL | Get,Set | Attributwert. |
| Test | BOOL | Get,Set | Attributwert. |
| eSource | E_AcsiQualitySource [▶ 424] | Get,Set | Attributwert. |
| OperatorBlocked | BOOL | Get,Set | Attributwert. |
| nValue | WORD(0..8191) | Get,Set | Attributwert im TwinCAT SPS WORD-Format. |

6.3.1.1.94 FB_AcsiDATypeRange

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBATypeENUMERATED](#) [[▶ 98](#)] -> [FB_AcsiDATypeRange](#)

FUNCTION_BLOCK FB_AcsiDATypeRange EXTENDS FB_AcsiBATypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|---------------------------------------|---------|---------------|
| eValue | E_AcsiRange [▶ 425] | Get,Set | Attributwert. |

6.3.1.1.95 FB_AcsiDATypeRangeConfig

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiDATypeRangeConfig](#)

FUNCTION_BLOCK FB_AcsiDATypeRangeConfig EXTENDS FB_AcsiCommonAttributeClass

 **Eingänge**

| Name | Typ | Beschreibung |
|-------|--|----------------|
| hhLim | FB_AcsiDATypeAnalogueValue [▶ 105] | Datenattribut. |
| hLim | FB_AcsiDATypeAnalogueValue [▶ 105] | Datenattribut. |
| lLim | FB_AcsiDATypeAnalogueValue [▶ 105] | Datenattribut. |
| llLim | FB_AcsiDATypeAnalogueValue [▶ 105] | Datenattribut. |
| min_ | FB_AcsiDATypeAnalogueValue [▶ 105] | Datenattribut. |
| max_ | FB_AcsiDATypeAnalogueValue [▶ 105] | Datenattribut. |
| limDb | FB_AcsiBATypeINT32U [▶ 101] | Datenattribut. |

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--|------------------|---------|------------------------|
| bPresence
[▶ 313] | BOOL | Get,Set | Attribut-Sichtbarkeit. |
| Fc [▶ 313] | DWORD | Get,Set | Funktionale Gruppe. |
| fHHLim | REAL | Get,Set | Attributwert. |
| fHLim | REAL | Get,Set | Attributwert. |
| fLLim | REAL | Get,Set | Attributwert. |
| fLLLim | REAL | Get,Set | Attributwert. |
| fMax | REAL | Get,Set | Attributwert. |
| fMin | REAL | Get,Set | Attributwert. |
| nLimDb | DWORD(0..100000) | Get,Set | Attributwert. |

6.3.1.1.96 FB_AcsiDATypeRcdMod

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBATypeENUMERATED](#) [[▶ 98](#)] -> [FB_AcsiDATypeRcdMod](#)

```
FUNCTION_BLOCK FB_AcsiDATypeRcdMod EXTENDS FB_AcsiBATypeENUMERATED
```

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiRcdMod [▶ 425] | Get,Set | Attributwert. |

6.3.1.1.97 FB_AcsiDATypeReasonCode

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)]-> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiDATypeReasonCode](#)

```
FUNCTION_BLOCK FB_AcsiDATypeReasonCode EXTENDS FB_AcsiCommonAttributeClass
```


 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|----------------------|---|---------|--|
| cValue | ST_AcsiReasonCode [▶ 465] | Get,Set | Attributwert als strukturierter Typ. |
| DataChange | BOOL | Get,Set | Attributwert. |
| QualityChange | BOOL | Get,Set | Attributwert. |
| DataUpdate | BOOL | Get,Set | Attributwert. |
| Integrity | BOOL | Get,Set | Attributwert. |
| GeneralInterrogation | BOOL | Get,Set | Attributwert. |
| ApplicationTrigger | BOOL | Get,Set | Attributwert. |
| nValue | BYTE(0..127) | Get,Set | Attributwert im TwinCAT SPS BYTE-Format. |

6.3.1.1.98 FB_AcsiDATypeReTrMod

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDATypeReTrMod](#)

FUNCTION_BLOCK FB_AcsiDATypeReTrMod EXTENDS FB_AcsiBATypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|---------------------------------------|---------|---------------|
| eValue | E_AcsiReTrMod [▶ 426] | Get,Set | Attributwert. |

6.3.1.1.99 FB_AcsiDATypeRotDir

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDATypeRotDir](#)

FUNCTION_BLOCK FB_AcsiDATypeRotDir EXTENDS FB_AcsiBATypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|--------------------------------------|---------|---------------|
| eValue | E_AcsiRotDir [▶ 426] | Get,Set | Attributwert. |

6.3.1.1.100 FB_AcsiDATypeRstMod

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDATypeRstMod](#)

```
FUNCTION_BLOCK FB_AcsiDATypeRstMod EXTENDS FB_AcsiBATypeENUMERATED
```

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|--------------------------------------|---------|---------------|
| eValue | E_AcsiRstMod [▶ 427] | Get,Set | Attributwert. |

6.3.1.1.101 FB_AcsiDATypeSboClasses

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDATypeSboClasses](#)

```
FUNCTION_BLOCK FB_AcsiDATypeSboClasses EXTENDS FB_AcsiBATypeENUMERATED
```

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiSboClasses [▶ 427] | Get,Set | Attributwert. |

6.3.1.1.102 FB_AcsiDATypeScaledValueConfig

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiDATypeScaledValueConfig](#)

```
FUNCTION_BLOCK FB_AcsiDATypeScaledValueConfig EXTENDS FB_AcsiCommonAttributeClass IMPLEMENTS I_AcsiDATypeScaledValueConfig
```

Schnittstellen

| Typ | Beschreibung |
|---|---|
| I_AcsiDATypeScaledValueConfig [▶ 308] | Schnittstellenzeiger der ScaledValueConfig-Objektinstanz. |

Eingänge

| Name | Typ | Beschreibung |
|-------------|---|----------------|
| scaleFactor | FB_AcsiBATypeFLOAT32 [▶ 99] | Datenattribut. |
| offset | FB_AcsiBATypeFLOAT32 [▶ 99] | Datenattribut. |

Methoden

| Name | Beschreibung |
|-----------------------------------|---|
| ScaleAToP [▶ 308] | Skaliert einen Attributwert in einen Prozesswert. |
| ScalePToA [▶ 309] | Skaliert einen Prozesswert in einen Attributwert. |

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|-------------------------------------|-------|---------|-----------------------------------|
| bPresence [▶ 313] | BOOL | Get,Set | Attribut-Sichtbarkeit. |
| Fc [▶ 313] | DWORD | Get,Set | Funktionale Gruppe. |
| fOffset | REAL | Get,Set | Attributwert: Skallierungsoffset. |
| fScaleFactor | REAL | Get,Set | Attributwert: Skalierungsfaktor. |

6.3.1.1.103 FB_AcsiDATypeSeqT

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBATypeENUMERATED](#) [[▶ 98](#)] -> [FB_AcsiDATypeSeqT](#)

FUNCTION_BLOCK FB_AcsiDATypeSeqT EXTENDS FB_AcsiBATypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiSeqT [▶ 428] | Get,Set | Attributwert. |

6.3.1.1.104 FB_AcsiDATypeServiceError

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBATypeENUMERATED](#) [[▶ 98](#)] -> [FB_AcsiDATypeServiceError](#)

FUNCTION_BLOCK FB_AcsiDATypeServiceError EXTENDS FB_AcsiBATypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiServiceError [▶ 428] | Get,Set | Attributwert. |

6.3.1.1.105 FB_AcsiDATypeSetCharact

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)]-> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBATypeENUMERATED](#) [[▶ 98](#)] -> [FB_AcsiDATypeSetCharact](#)

FUNCTION_BLOCK FB_AcsiDATypeSetCharact EXTENDS FB_AcsiBATypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|-----------------------------|---------|---------------|
| eValue | E_AcsiSetCharact
[▶ 430] | Get,Set | Attributwert. |

6.3.1.1.106 FB_AcsiDATypeSev

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [▶ 154] -> [FB_AcsiCommonDataClass](#) [▶ 148] -> [FB_AcsiCommonAttributeClass](#) [▶ 146] -> [FB_AcsiBTypeENUMERATED](#) [▶ 98] -> [FB_AcsiDATypeSev](#)

```
FUNCTION_BLOCK FB_AcsiDATypeSev EXTENDS FB_AcsiBTypeENUMERATED
```

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|-------------------|---------|---------------|
| eValue | E_AcsiSev [▶ 431] | Get,Set | Attributwert. |

6.3.1.1.107 FB_AcsiDATypeShOpCap

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [▶ 154] -> [FB_AcsiCommonDataClass](#) [▶ 148] -> [FB_AcsiCommonAttributeClass](#) [▶ 146] -> [FB_AcsiBTypeENUMERATED](#) [▶ 98] -> [FB_AcsiDATypeShOpCap](#)

```
FUNCTION_BLOCK FB_AcsiDATypeShOpCap EXTENDS FB_AcsiBTypeENUMERATED
```

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|-----------------------|---------|---------------|
| eValue | E_AcsiShOpCap [▶ 431] | Get,Set | Attributwert. |

6.3.1.1.108 FB_AcsiDATypeSIUnit

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [▶ 154] -> [FB_AcsiCommonDataClass](#) [▶ 148] -> [FB_AcsiCommonAttributeClass](#) [▶ 146] -> [FB_AcsiBTypeENUMERATED](#) [▶ 98] -> [FB_AcsiDATypeSIUnit](#)

```
FUNCTION_BLOCK FB_AcsiDATypeSIUnit EXTENDS FB_AcsiBTypeENUMERATED
```

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|----------------------|---------|---------------|
| eValue | E_AcsiSIUnit [▶ 432] | Get,Set | Attributwert. |

6.3.1.1.109 FB_AcsiDATypeSptEndSt

Namensraum: [Tc3_Acsi](#) [[96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[154](#)] -> [FB_AcsiCommonDataClass](#) [[148](#)] -> [FB_AcsiCommonAttributeClass](#) [[146](#)] -> [FB_AcsiBATypeENUMERATED](#) [[98](#)] -> [FB_AcsiDATypeSptEndSt](#)

FUNCTION_BLOCK FB_AcsiDATypeSptEndSt EXTENDS FB_AcsiBATypeENUMERATED



Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiSptEndSt [434] | Get,Set | Attributwert. |

6.3.1.1.110 FB_AcsiDATypeStClcTun

Namensraum: [Tc3_Acsi](#) [[96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[154](#)] -> [FB_AcsiCommonDataClass](#) [[148](#)] -> [FB_AcsiCommonAttributeClass](#) [[146](#)] -> [FB_AcsiBATypeENUMERATED](#) [[98](#)] -> [FB_AcsiDATypeStClcTun](#)

FUNCTION_BLOCK FB_AcsiDATypeStClcTun EXTENDS FB_AcsiBATypeENUMERATED



Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiStClcTun [435] | Get,Set | Attributwert. |

6.3.1.1.111 FB_AcsiDATypeStrWeekDay

Namensraum: [Tc3_Acsi](#) [[96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[154](#)] -> [FB_AcsiCommonDataClass](#) [[148](#)] -> [FB_AcsiCommonAttributeClass](#) [[146](#)] -> [FB_AcsiBATypeENUMERATED](#) [[98](#)] -> [FB_AcsiDATypeStrWeekDay](#)

FUNCTION_BLOCK FB_AcsiDATypeStrWeekDay EXTENDS FB_AcsiBATypeENUMERATED



Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiStrWeekDay [436] | Get,Set | Attributwert. |

6.3.1.1.112 FB_AcsiDATypeSwOpCap

Namensraum: [Tc3_Acsi](#) [[96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDTypeSwOpCap](#)

FUNCTION_BLOCK FB_AcsiDTypeSwOpCap EXTENDS FB_AcsiBTypeENUMERATED

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|---------------------------------------|---------|---------------|
| eValue | E_AcsiSwOpCap [▶ 436] | Get,Set | Attributwert. |

6.3.1.1.113 FB_AcsiDTypeSwTyp

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDTypeSwTyp](#)

FUNCTION_BLOCK FB_AcsiDTypeSwTyp EXTENDS FB_AcsiBTypeENUMERATED

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|-------------------------------------|---------|---------------|
| eValue | E_AcsiSwTyp [▶ 437] | Get,Set | Attributwert. |

6.3.1.1.114 FB_AcsiDTypeTcmd

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeCODEDENUM \[▶ 98\]](#) -> [FB_AcsiDTypeTcmd](#)

FUNCTION_BLOCK FB_AcsiDTypeTcmd EXTENDS FB_AcsiBTypeCODEDENUM

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|------------------------------------|---------|--|
| eValue | E_AcsiTcmd [▶ 437] | Get,Set | Attributwert als Aufzählungstyp. |
| b0 | BOOL | Get,Set | Attributwert. |
| b1 | BOOL | Get,Set | Attributwert. |
| nValue | BYTE(0..3) | Get,Set | Attributwert im TwinCAT SPS BYTE-Format. |

6.3.1.1.115 FB_AcsiDTypeTimeStamp

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiDTypeTimeStamp](#)

FUNCTION_BLOCK FB_AcsiDTypeTimeStamp EXTENDS FB_AcsiCommonAttributeClass

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|----------------------|---|---------|---|
| tValue | T.UtcTime [▶ 479] | Get,Set | Attributwert: Zeitinformation als strukturierter Typ. |
| SecondSinceEpoch | DT | Get,Set | Attributwert. |
| FractionOfSecond | T.UINT24 [▶ 479] | Get,Set | Attributwert. |
| Quality | T.UtcTimeQuality [▶ 480] | Get,Set | Attributwert: Quality als strukturierter Typ. |
| LeapSecondsKnown | BOOL | Get,Set | Attributwert. |
| ClockFailure | BOOL | Get,Set | Attributwert. |
| ClockNotSynchronized | BOOL | Get,Set | Attributwert. |
| eAccuracy | E.UtcTimeAccuracy [▶ 449] | Get,Set | Attributwert. |
| sValue | STRING(39) | Get,Set | Attributwert: Zeitinformation als String mit folgendem Format:
„UT#1970-01-01-00:00:00.000000000 000 0“. |
| nValue | LWORD | Get,Set | Attributwert: Zeitinformation im TwinCAT SPS LWORD-Format. |

6.3.1.1.116 FB_AcsiDATypeTmSyn

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBATypeENUMERATED](#) [[▶ 98](#)] -> [FB_AcsiDATypeTmSyn](#)

FUNCTION_BLOCK FB_AcsiDATypeTmSyn EXTENDS FB_AcsiBATypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|---------------------------------------|---------|---------------|
| eValue | E.AcsiTmSyn [▶ 438] | Get,Set | Attributwert. |

6.3.1.1.117 FB_AcsiDATypeTnkTyp

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBATypeENUMERATED](#) [[▶ 98](#)] -> [FB_AcsiDATypeTnkTyp](#)

FUNCTION_BLOCK FB_AcsiDATypeTnkTyp EXTENDS FB_AcsiBATypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E.AcsiTnkTyp [▶ 438] | Get,Set | Attributwert. |

6.3.1.1.118 FB_AcsiDATypeTpcRxMod

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDATypeTpcRxMod](#)

FUNCTION_BLOCK FB_AcsiDATypeTpcRxMod EXTENDS FB_AcsiBATypeENUMERATED

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiTpcRxMod [▶ 438] | Get,Set | Attributwert. |

6.3.1.1.119 FB_AcsiDATypeTpcTxMod

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDATypeTpcTxMod](#)

FUNCTION_BLOCK FB_AcsiDATypeTpcTxMod EXTENDS FB_AcsiBATypeENUMERATED

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiTpcTxMod [▶ 439] | Get,Set | Attributwert. |

6.3.1.1.120 FB_AcsiDATypeTrBeh

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDATypeTrBeh](#)

FUNCTION_BLOCK FB_AcsiDATypeTrBeh EXTENDS FB_AcsiBATypeENUMERATED

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|-------------------------------------|---------|---------------|
| eValue | E_AcsiTrBeh [▶ 439] | Get,Set | Attributwert. |

6.3.1.1.121 FB_AcsiDATypeTrgMod

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDTypeTrgMod](#)

FUNCTION_BLOCK FB_AcsiDTypeTrgMod EXTENDS FB_AcsiBTypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|--------------------------------------|---------|---------------|
| eValue | E_AcsiTrgMod [▶ 440] | Get,Set | Attributwert. |

6.3.1.1.122 FB_AcsiDTypeTriggerConditions

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiDTypeTriggerConditions](#)

FUNCTION_BLOCK FB_AcsiDTypeTriggerConditions EXTENDS FB_AcsiCommonAttributeClass

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|----------------------|--|---------|--|
| cValue | ST_AcsiTriggerConditions [▶ 466] | Get,Set | Attributwert als strukturierter Typ. |
| DataChange | BOOL | Get,Set | Attributwert. |
| QualityChange | BOOL | Get,Set | Attributwert. |
| DataUpdate | BOOL | Get,Set | Attributwert. |
| Integrity | BOOL | Get,Set | Attributwert. |
| GeneralInterrogation | BOOL | Get,Set | Attributwert. |
| nValue | BYTE(0..63) | Get,Set | Attributwert im TwinCAT SPS BYTE-Format. |

6.3.1.1.123 FB_AcsiDTypeTrMod

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDTypeTrMod](#)

FUNCTION_BLOCK FB_AcsiDTypeTrMod EXTENDS FB_AcsiBTypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|-------------------------------------|---------|---------------|
| eValue | E_AcsiTrMod [▶ 440] | Get,Set | Attributwert. |

6.3.1.1.124 FB_AcsiDTypeTypRsCrv

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDATypeTypRsCrv](#)

```
FUNCTION_BLOCK FB_AcsiDATypeTypRsCrv EXTENDS FB_AcsiBATypeENUMERATED
```



Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiTypRsCrv [▶ 440] | Get,Set | Attributwert. |

6.3.1.1.125 FB_AcsiDATypeUnbDetMth

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDATypeUnbDetMth](#)

```
FUNCTION_BLOCK FB_AcsiDATypeUnbDetMth EXTENDS FB_AcsiBATypeENUMERATED
```



Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|---|---------|---------------|
| eValue | E_AcsiUnbDetMth [▶ 441] | Get,Set | Attributwert. |

6.3.1.1.126 FB_AcsiDATypeUnBlkMod

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDATypeUnBlkMod](#)

```
FUNCTION_BLOCK FB_AcsiDATypeUnBlkMod EXTENDS FB_AcsiBATypeENUMERATED
```



Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiUnBlkMod [▶ 441] | Get,Set | Attributwert. |

6.3.1.1.127 FB_AcsiDATypeUnicode255

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeUNICODESTRING \[▶ 103\]](#) -> [FB_AcsiDATypeUnicode255](#)

```
FUNCTION_BLOCK FB_AcsiBATypeUnicode255 EXTENDS FB_AcsiBATypeUNICODESTRING
```

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|--------------|---------|---------------|
| sValue | WSTRING(255) | Get,Set | Attributwert. |

6.3.1.1.128 FB_AcsiDATypeUnit

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiDATypeUnit](#)

FUNCTION_BLOCK FB_AcsiDATypeUnit EXTENDS FB_AcsiCommonAttributeClass IMPLEMENTS I_AcsiDATypeUnit

 **Schnittstellen**

| Typ | Beschreibung |
|--|--|
| I_AcsiDATypeUnit [▶ 309] | Schnittstellenzeiger der Unit-Objektinstanz. |

 **Eingänge**

| Name | Typ | Beschreibung |
|------------|---|----------------|
| SIUnit | FB_AcsiDATypeSIUnit [▶ 136] | Datenattribut. |
| multiplier | FB_AcsiDATypeMultiplier [▶ 120] | Datenattribut. |

 **Methoden**

| Name | Beschreibung |
|-----------------------------------|---|
| ScaleAToP [▶ 309] | Skaliert einen Attributwert in einen Prozesswert. |
| ScalePToA [▶ 309] | Skaliert einen Prozesswert in einen Attributwert. |

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|-----------------------------------|--|---------|------------------------------|
| bPresence [▶ 313] | BOOL | Get,Set | Attribut-Sichtbarkeit. |
| Fc [▶ 313] | DWORD | Get,Set | Funktionale Gruppe. |
| eMultiplier | E_AcsiMultiplier [▶ 416] | Get,Set | Attributwert: Multiplikator. |
| eSIUnit | E_AcsiSIUnit [▶ 432] | Get,Set | Attributwert: SI-Einheit. |

6.3.1.1.129 FB_AcsiDATypeValWithTrans

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiDATypeValWithTrans](#)

```
FUNCTION_BLOCK FB_AcsiDATypeValWithTrans EXTENDS FB_AcsiCommonAttributeClass
```

Eingänge

| Name | Typ | Beschreibung |
|----------|--|----------------|
| posVal | FB_AcsiBTypeINT8 [▶ 99] | Datenattribut. |
| transInd | FB_AcsiBTypeBOOLEAN [▶ 98] | Datenattribut. |

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|-----------------------------------|---------------|---------|------------------------|
| bPresence [▶ 313] | BOOL | Get,Set | Attribut-Sichtbarkeit. |
| Fc [▶ 313] | DWORD | Get,Set | Funktionale Gruppe. |
| iPosVal | SINT(-64..63) | Get,Set | Attributwert. |
| bTransInd | BOOL | Get,Set | Attributwert. |

6.3.1.1.130 FB_AcsiDATypeVector

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiDATypeVector](#)

```
FUNCTION_BLOCK FB_AcsiDATypeVector EXTENDS FB_AcsiCommonAttributeClass
```

Eingänge

| Name | Typ | Beschreibung |
|------|--|----------------|
| mag | FB_AcsiDATypeAnalogueValue [▶ 105] | Datenattribut. |
| ang | FB_AcsiDATypeAnalogueValue [▶ 105] | Datenattribut. |

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|-----------------------------------|-------|---------|------------------------|
| bPresence [▶ 313] | BOOL | Get,Set | Attribut-Sichtbarkeit. |
| Fc [▶ 313] | DWORD | Get,Set | Funktionale Gruppe. |
| fMag | REAL | Get,Set | Attributwert. |
| iMag | DINT | Get,Set | Attributwert. |
| fAng | REAL | Get,Set | Attributwert. |
| iAng | DINT | Get,Set | Attributwert. |

6.3.1.1.131 FB_AcsiDATypeVisString129

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeVISIBLESTRING \[▶ 103\]](#) -> [FB_AcsiDATypeVisString129](#)

FUNCTION_BLOCK FB_AcsiDATypeVisString129 EXTENDS FB_AcsiBATypeVISIBLESTRING

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|-------------|---------|---------------|
| sValue | STRING(129) | Get,Set | Attributwert. |

6.3.1.1.132 FB_AcsiDATypeVisString255

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeVISIBLESTRING \[▶ 103\]](#) -> [FB_AcsiDATypeVisString255](#)

FUNCTION_BLOCK FB_AcsiDATypeVisString255 EXTENDS FB_AcsiBATypeVISIBLESTRING

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|-------------|---------|---------------|
| sValue | STRING(255) | Get,Set | Attributwert. |

6.3.1.1.133 FB_AcsiDATypeVisString64

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeVISIBLESTRING \[▶ 103\]](#) -> [FB_AcsiDATypeVisString64](#)

FUNCTION_BLOCK FB_AcsiDATypeVisString64 EXTENDS FB_AcsiBATypeVISIBLESTRING

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|------------|---------|---------------|
| sValue | STRING(64) | Get,Set | Attributwert. |

6.3.1.1.134 FB_AcsiDATypeWeekDay

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBATypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiDATypeWeekDay](#)

FUNCTION_BLOCK FB_AcsiDATypeWeekDay EXTENDS FB_AcsiBATypeENUMERATED

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|---|---------|----------------------------------|
| eValue | E_AcsiWeekDay [▶ 442] | Get,Set | Attributwert als Aufzählungstyp. |

6.3.1.1.135 FB_AcsiDATypeWeiMod

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBTypeENUMERATED](#) [[▶ 98](#)] -> [FB_AcsiDATypeWeiMod](#)

```
FUNCTION_BLOCK FB_AcsiDATypeWeiMod EXTENDS FB_AcsiBTypeENUMERATED
```

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|--|---------|---------------|
| eValue | E_AcsiWeiMod [▶ 443] | Get,Set | Attributwert. |

6.3.1.1.136 FB_AcsiDATypeDatSetRef

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#) [[▶ 146](#)] -> [FB_AcsiBTypeVISIBLESTRING](#) [[▶ 103](#)] -> [FB_AcsiDATypeObjRef](#) [[▶ 120](#)] -> [FB_AcsiDATypeDatSetRef](#)

```
FUNCTION_BLOCK FB_AcsiDATypeDatSetRef EXTENDS FB_AcsiDATypeObjRef
```

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|----------|--|---------|-----------------------------|
| ipDatSet | I_AcsiCommonDataSetClass [▶ 313] | Get | Das referenzierte Data-Set. |

6.3.1.2 FB_AcsiCommonAttributeClass

Der Funktionsbaustein [FB_AcsiCommonAttributeClass](#) gehört zu den Grundbausteinen des TwinCAT IEC 61850 Datenmodells. Jede vom TwinCAT Telecontrol Configurator generierte Datenattributinstanz (Data Attribute, DA) ist von [FB_AcsiCommonAttributeClass](#) abgeleitet/erweitert worden.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCommonAttributeClass](#)

```
FUNCTION_BLOCK FB_AcsiCommonAttributeClass EXTENDS FB_AcsiCommonDataClass IMPLEMENTS I_AcsiCommonAttributeClass
```

 Schnittstellen

| Typ | Beschreibung |
|--|---|
| I_AcsiCommonAttributeClass [▶ 310] | Schnittstelle der Datenattributinstanz. |

 Methoden

| Name | Beschreibung |
|---|---|
| AddAttributeToContainer [▶ 310] | Fügt ein Datenattributobjekt dem Container eines anderen Datenattributobjekts hinzu |

Eigenschaften

-

6.3.1.3 FB_AcsiCommonBufferedReportControlBlockClass

Der Funktionsbaustein FB_AcsiCommonBufferedReportControlClass gehört zu den Grundbausteinen des TwinCAT IEC 61850 Datenmodells. Jede vom TwinCAT Telecontrol Configurator generierte Berichtssteuerblockinstanz für gepufferte Reports (Buffered Report Control Block, BRCB) ist von FB_AcsiCommonBufferedReportControlBlockClass abgeleitet/erweitert worden.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -

> [FB_AcsiCommonControlBlockClass \[▶ 147\]](#) -> **FB_AcsiCommonBufferedReportControlBlockClass**

```
FUNCTION_BLOCK FB_AcsiCommonBufferedReportControlBlockClass EXTENDS FB_AcsiCommonControlBlockClass IMPLEMENTS I_AcsiCommonBufferedReportControlBlockClass
```

 Schnittstellen

| Typ | Beschreibung |
|--|---|
| I_AcsiCommonBufferedReportControlClass [▶ 310] | Schnittstelle der Berichtssteuerblockinstanz. |

Methoden

-

Eigenschaften

-

6.3.1.4 FB_AcsiCommonControlBlockClass

Der Funktionsbaustein FB_AcsiCommonControlBlockClass gehört zu den Grundbausteinen des TwinCAT IEC 61850 Datenmodells. Der Funktionsbaustein dient als Basis für alle Steuerblock-Funktionsbausteine.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -

> **FB_AcsiCommonControlBlockClass**

```
FUNCTION_BLOCK FB_AcsiCommonControlBlockClass EXTENDS FB_AcsiCommonDataClass IMPLEMENTS I_AcsiCommonControlBlockClass
```

Schnittstellen

| Typ | Beschreibung |
|---|---|
| I_AcsiCommonControlBlockClass ▶ 311 | Schnittstelle des allgemeinen Steuerblocks. |

6.3.1.5 FB_AcsiCommonControllableDataClass

Der Funktionsbaustein FB_AcsiCommonControllableDataClass gehört zu den Grundbausteinen des TwinCAT IEC 61850 Datenmodells. Jede vom TwinCAT Telecontrol Configurator generierte Instanz der Datenklasse für die Schaltersteuerung (Common Data Class, CDC: SPC, DPC, INC, ...) ist von FB_AcsiCommonControllableDataClass abgeleitet/erweitert worden.

Namensraum: [Tc3_Acsi](#) [▶ 96](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [▶ 154](#) -> [FB_AcsiCommonDataClass](#) [▶ 148](#) -
> [FB_AcsiCommonControllableDataClass](#)

```
FUNCTION_BLOCK FB_AcsiCommonControllableDataClass EXTENDS FB_AcsiCommonDataClass IMPLEMENTS I_AcsiCommonControllableDataClass
```

Schnittstellen

| Typ | Beschreibung |
|---|--|
| I_AcsiCommonControllableDataClass ▶ 311 | Schnittstelle der Objektinstanz für die Schaltersteuerung. |

Methoden

-

Eigenschaften

-

6.3.1.6 FB_AcsiCommonDataClass

Der Funktionsbaustein FB_AcsiCommonDataClass gehört zu den Grundbausteinen des TwinCAT IEC 61850 Datenmodells. Jede vom TwinCAT Telecontrol Configurator generierte Datenobjektinstanz (DO) ist von FB_AcsiCommonDataClass abgeleitet/erweitert worden.

Namensraum: [Tc3_Acsi](#) [▶ 96](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [▶ 154](#) -> [FB_AcsiCommonDataClass](#)

```
FUNCTION_BLOCK FB_AcsiCommonDataClass EXTENDS FB_AcsiCommonNodeClass IMPLEMENTS I_AcsiCommonDataClass
```

Schnittstellen

| Typ | Beschreibung |
|---|---------------------------------------|
| I_AcsiCommonDataClass ▶ 311 | Schnittstelle der Datenobjektinstanz. |

 **Methoden**

| Name | Beschreibung |
|---|---|
| AddAttributeToContainer [▶ 312] | Fügt ein Datenattributobjekt dem Container eines Datenobjekts hinzu |
| AddDataToContainer [▶ 312] | Fügt ein Datenobjekt dem Container eines anderen Datenobjekts hinzu |
| AddToDataSet [▶ 312] | Fügt ein Datenobjekt dem Membercontainer des Datensatzes hinzu |
| DeleteFromDataSet [▶ 313] | Entfernt ein Datenobjekt aus dem Membercontainer des Datensatzes |

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|-----------------------------------|-------|---------|------------------------|
| bPresence [▶ 313] | BOOL | Get,Set | Attribut-Sichtbarkeit |
| Fc [▶ 313] | DWORD | Get,Set | Funktionale Gruppe |
| TrgOp [▶ 313] | DWORD | Get,Set | Auslöser eines Reports |

6.3.1.7 FB_AcsiCommonDataSetClass

Der Funktionsbaustein FB_AcsiCommonDataSetClass gehört zu den Grundbausteinen des TwinCAT IEC 61850 Datenmodells. Jede vom TwinCAT Telecontrol Configurator generierte Datensatzinstanz (Data-Set) ist von FB_AcsiCommonDataSetClass abgeleitet/erweitert worden.

Namensraum: Tc3_Acsi [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataSetClass](#)

```
FUNCTION_BLOCK FB_AcsiCommonDataSetClass EXTENDS FB_AcsiCommonNodeClass IMPLEMENTS I_AcsiCommonDataS
etClass
VAR_OUTPUT
    stInfo : ST_AcsiDataSetClassInfo;
END_VAR
```

 **Schnittstellen**

| Typ | Beschreibung |
|--|-------------------------------------|
| I_AcsiCommonDataSetClass [▶ 313] | Schnittstelle der Datensatzinstanz. |

 **Ausgänge**

| Name | Typ | Beschreibung |
|--------|---|---|
| stInfo | ST_AcsiDataSetClassInfo [▶ 456] | Status und Statistikdaten der Datensatzinstanz. |

 **Methoden**

| Name | Beschreibung |
|--|--|
| AddMember [▶ 314] | Fügt ein neues Datenobjekt dem Membercontainer hinzu |
| DeleteMember [▶ 314] | Entfernt ein Datenobjekt aus dem Membercontainer |
| DeleteAllMembers [▶ 315] | Entfernt alle Datenobjekte aus dem Membercontainer |

Methoden zur Ereignisbehandlung

| Name | Beschreibung |
|--------------------------------|--|
| OnInit [▶ 315] | Initialisiert/Konfiguriert die Datenobjekte im Membercontainer |

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|----------|-------|---------|--|
| nMembers | UDINT | Get | Anzahl der Datenobjekte im Membercontainer |

6.3.1.8 FB_AcsiCommonGooseControlBlockClass

Der Funktionsbaustein FB_AcsiCommonGooseControlBlockClass gehört zu den Grundbausteinen des TwinCAT IEC 61850 Datenmodells. Jede vom TwinCAT Telecontrol Configurator generierte GOOSE-Steuerblockinstanz ist von FB_AcsiCommonGooseControlBlockClass abgeleitet/erweitert worden.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -

> [FB_AcsiCommonControlBlockClass \[▶ 147\]](#) -> [FB_AcsiCommonGooseControlBlockClass](#)

```
FUNCTION_BLOCK FB_AcsiCommonGooseControlBlockClass EXTENDS FB_AcsiCommonControlBlockClass IMPLEMENTS
I_AcsiCommonGooseControlBlockClass
```

Schnittstellen

| Typ | Beschreibung |
|--|---|
| I_AcsiCommonGooseControlBlockClass [▶ 315] | Schnittstelle der GOOSE-Steuerblockinstanz. |

6.3.1.9 FB_AcsiCommonIEDGroupClass

Der Funktionsbaustein FB_AcsiCommonIEDGroupClass gehört zu den Grundbausteinen der TwinCAT IEC 61850 Datenmodells. Objekte, die von FB_AcsiCommonIEDGroupClass abgeleitet/erweitert wurden repräsentieren eine Gruppe von mehreren IEDs. Alle IEDs in dieser Gruppe verwenden für die Kommunikation den gleichen „Real-Time Ethernet Adapter (Multiple Protocol Handler)“.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonIEDGroupClass](#)

```
FUNCTION_BLOCK FB_AcsiCommonIEDGroupClass IMPLEMENTS I_AcsiCommonIEDGroupClass
VAR_OUTPUT
    stInfo : ST_AcsiIEDGroupClassInfo;
END_VAR
```

Schnittstellen

| Typ | Beschreibung |
|---|--|
| I_AcsiCommonIEDGroupClass [▶ 315] | Schnittstelle der Objektinstanz einer Gruppe von IEDs. |

 **Ausgänge**

| Name | Typ | Beschreibung |
|--------|--|---|
| stInfo | ST_AcsiIEDGroupClassInfo [▶ 457] | Status und Statistikdaten der IED-Gruppe. |

 **Methoden**

| Name | Beschreibung |
|---|--|
| AddIEDToContainer [▶ 151] | Fügt ein neues Gerät dem Container der IED-Gruppe hinzu. |

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|-------------|--|----------|--------------------------------|
| nIEDs | UDINT | Get | Anzahl der IEDs im Container. |
| sObjectName | T_AcsiObjectName [▶ 475] | Get, Set | Objektname. |
| Tag | T_AcsiTag [▶ 476] | Get, Set | Zusätzliche Objektinformation. |

6.3.1.9.1 AddIEDToContainer

Fügt ein neues IED dem Container der IED-Gruppe hinzu.

```
METHOD FINAL AddIEDToContainer : BOOL
VAR_INPUT
    ipIED      : I_AcsiCommonIntelligentElectronicDeviceClass;
END_VAR
VAR_OUTPUT
    bMatched  : BOOL;
    ipDirNode : I_AcsiIEDContainerClass;
END_VAR
```

ipIED: Schnittstellenzeiger vom Typ [I_AcsiCommonIntelligentElectronicDeviceClass](#) [[▶ 316](#)] des IED-Gerätes, welches dem Container hinzugefügt werden soll.

bMatched: Bei TRUE existiert das Objekt bereits im Container.

ipDirNode: Schnittstellenzeiger des neuen Container-Listenelements.

Rückgabeparameter: Positive Meldung (TRUE) beim Erfolg und negative Meldung (FALSE) bei einem Fehler.

6.3.1.10 FB_AcsiCommonIntelligentElectronicDeviceClass

Der Funktionsbaustein [FB_AcsiCommonIntelligentElectronicDeviceClass](#) gehört zu den Grundbausteinen des TwinCAT IEC 61850 Datenmodells. Jede vom TwinCAT Telecontrol Configurator generierte Serverinstanz (IED) ist von [FB_AcsiCommonIntelligentElectronicDeviceClass](#) abgeleitet/erweitert worden.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: [Tc3_Acsi](#) ([Tc3_Acsi.compiled-library](#))

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonIntelligentElectronicDeviceClass](#)

```
FUNCTION_BLOCK FB_AcsiCommonIntelligentElectronicDeviceClass EXTENDS FB_AcsiCommonNodeClass IMPLEMENTS
TS I_AcsiCommonIntelligentElectronicDeviceClass
VAR_OUTPUT
    stInfo : ST_AcsiIntelligentElectronicDeviceClassInfo;
END_VAR
```

Schnittstellen

| Typ | Beschreibung |
|--|---|
| I_AcsiCommonIntelligentElectronicDeviceClass [▶ 316] | Schnittstelle der Objektinstanz der obersten Ebene der Datenstruktur des Servers. |

Ausgänge

| Name | Typ | Beschreibung |
|--------|--|-------------------------------------|
| stInfo | ST_AcsiIntelligentElectronicDeviceClassInfo
[▶ 458] | Status und Statistikdaten des IEDs. |

Methoden

| Name | Beschreibung |
|---|--|
| AddLogicalDeviceToContainer [▶ 316] | Fügt ein neues Logisches Gerät dem Container des Servers hinzu |

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|-----------------|-------|---------|--|
| nLogicalDevices | UDINT | Get | Anzahl der Logischen Geräte im Container des Servers |

6.3.1.11 FB_AcsiCommonLogicalDeviceClass

Der Funktionsbaustein FB_AcsiCommonLogicalDeviceClass gehört zu den Grundbausteinen des TwinCAT IEC 61850 Datenmodells. Jede vom TwinCAT Telecontrol Configurator generierte Instanz des Logischen Gerätes (Logical Device, LD) ist von FB_AcsiCommonLogicalDeviceClass abgeleitet/erweitert worden.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonLogicalDeviceClass](#)

```
FUNCTION_BLOCK FB_AcsiCommonLogicalDeviceClass EXTENDS FB_AcsiCommonNodeClass IMPLEMENTS I_AcsiCommonLogicalDeviceClass
VAR_OUTPUT
    stInfo : ST_AcsiLogicalDeviceClassInfo;
END_VAR
```

Schnittstellen

| Typ | Beschreibung |
|--|--|
| I_AcsiCommonLogicalDeviceClass [▶ 317] | Schnittstelle der Objektinstanz des Logischen Gerätes. |

Ausgänge

| Name | Typ | Beschreibung |
|--------|--|--|
| stInfo | ST_AcsiLogicalDeviceClassInfo
[▶ 458] | Status und Statistikdaten des Logischen-Gerätes. |

 **Methoden**

| Name | Beschreibung |
|---|---|
| AddLogicalNodeToContainer [▶ 317] | Fügt einen neuen Logischen Knoten hinzu |

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|---------------|-------|---------|--|
| nLogicalNodes | UDINT | Get | Anzahl der Logischen Knoten im Container des Logischen Gerätes |

6.3.1.12 FB_AcsiCommonLogicalNodeClass

Der Funktionsbaustein FB_AcsiCommonLogicalNodeClass gehört zu den Grundbausteinen des TwinCAT IEC 61850 Datenmodells. Jede vom TwinCAT Telecontrol Configurator generierte Instanz des Logischen Knoten (Logical Node, LN) ist von FB_AcsiCommonLogicalNodeClass abgeleitet/erweitert worden.

Namensraum: [Tc3_Acsi](#) [\[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [\[▶ 154\]](#) -> FB_AcsiCommonLogicalNodeClass

```
FUNCTION_BLOCK FB_AcsiCommonLogicalNodeClass EXTENDS FB_AcsiCommonNodeClass IMPLEMENTS I_AcsiCommonLogicalNodeClass
VAR_OUTPUT
    stInfo : ST_AcsiLogicalNodeClassInfo;
END_VAR
```

 **Schnittstellen**

| Typ | Beschreibung |
|--|--|
| I_AcsiCommonLogicalNodeClass [▶ 317] | Schnittstelle der Objektinstanz des Logischen Knotens. |

 **Ausgänge**

| Name | Typ | Beschreibung |
|--------|---|---|
| stInfo | ST_AcsiLogicalNodeClassInfo [▶ 458] | Status und Statistikdaten des Logischen-Knoten. |

 **Methoden**

| Name | Beschreibung |
|--|--|
| AddDataToContainer [▶ 318] | Fügt ein neues Datenobjekt hinzu |
| AddDataSetToContainer [▶ 318] | Fügt einen neuen Datensatz hinzu |
| AddUnbufferedReportControlBlockToContainer [▶ 319] | Fügt eine neue Berichtssteuerblockinstanz für nicht gepufferte Reports hinzu |
| AddBufferedReportControlBlockToContainer [▶ 319] | Fügt eine neue Berichtssteuerblockinstanz für gepufferte Reports hinzu |
| AddGooseControlBlockToContainer [▶ 319] | Fügt eine neue Goose-Steuerblockinstanz hinzu |

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------------------------------|-------|---------|--|
| nDataSets | UDINT | Get | Anzahl der Datensätze im Container des Logischen Knoten |
| nBufferedReportControlBlocks | UDINT | Get | Anzahl der Berichtssteuerblockinstanzen für nicht gepufferte Reports |
| nUnbufferedReportControlBlocks | UDINT | Get | Anzahl der Berichtssteuerblockinstanzen für gepufferte Reports |
| nGooseControlBlocks | UDINT | Get | Anzahl der Goose-Steuerblockinstanzen |

6.3.1.13 FB_AcsiCommonNodeClass

Der Funktionsbaustein FB_AcsiCommonNodeClass gehört zu den Grundbausteinen des TwinCAT IEC 61850 Datenmodells. Jede vom TwinCAT Telecontrol Configurator generierte Objektinstanz ist von der Basis FB_AcsiCommonNodeClass abgeleitet/erweitert worden.

Namensraum: Tc3_Acsi [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

FB_AcsiCommonNodeClass

FUNCTION_BLOCK FB_AcsiCommonNodeClass IMPLEMENTS I_AcsiCommonNodeClass

 **Schnittstellen**

| Typ | Beschreibung |
|---|--|
| I_AcsiCommonNodeClass [▶ 320] | Schnittstelle der Basis-Knoteninstanz. |

 **Methoden**

| Name | Beschreibung |
|---|--|
| GetObjectReference
[▶ 320] | Liefert den Referenzpfad einer Objektinstanz |
| GetServerObject [▶ 321] | Liefert den Schnittstellenzeiger der Objektinstanz des Servers |

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--|---|---------|--------------|
| sObjectName
[▶ 321] | T_AcsiObjectName
[▶ 475] | Get,Set | Objektname |

6.3.1.14 FB_AcsiCommonUnbufferedReportControlBlockClass

Der Funktionsbaustein FB_AcsiCommonUnbufferedReportControlBlockClass gehört zu den Grundbausteinen des TwinCAT IEC 61850 Datenmodells. Jede vom TwinCAT Telecontrol Configurator generierte Berichtssteuerblockinstanz für ungepufferte Reports (Unbuffered Report Control Block, URCB) ist von FB_AcsiCommonUnbufferedReportControlBlockClass abgeleitet/erweitert worden.

Namensraum: Tc3_Acsi [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

FB_AcsiCommonNodeClass [▶ 154] -> FB_AcsiCommonDataClass [▶ 148] -
 > FB_AcsiCommonControlBlockClass [▶ 147] -> FB_AcsiCommonUnbufferedReportControlBlockClass

```
FUNCTION_BLOCK FB_AcsiCommonUnbufferedReportControlBlockClass EXTENDS FB_AcsiCommonControlBlockClass
    IMPLEMENTS I_AcsiCommonUnbufferedReportControlBlockClass
```

 **Schnittstellen**

| Typ | Beschreibung |
|---|---|
| I_AcsiCommonUnbufferedReportControlBlockClass [▶ 321] | Schnittstelle der Berichtssteuerblockinstanz. |

Methoden

-

Eigenschaften

-

6.3.1.15 Schaltersteuerung

6.3.1.15.1 FB_AcsiCDC_Cancel_AnalogueValue

Verwendung: APC.

Namensraum: Tc3_Acsi [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

FB_AcsiCommonNodeClass [▶ 154] -> FB_AcsiCommonDataClass [▶ 148] -
 > FB_AcsiCDC_Cancel_AnalogueValue

```
FUNCTION_BLOCK FB_AcsiCDC_Cancel_AnalogueValue EXTENDS FB_AcsiCommonDataClass
```

 **Eingänge**

| Name | Typ | Beschreibung |
|--------|------------------------------------|----------------|
| ctlVal | FB_AcsiDATypeAnalogueValue [▶ 105] | Datenattribut. |
| operTm | FB_AcsiDATypeTimeStamp [▶ 138] | Datenattribut. |
| origin | FB_AcsiDATypeOriginator [▶ 125] | Datenattribut. |
| ctlNum | FB_AcsiBATypeINT8U [▶ 100] | Datenattribut. |
| T | FB_AcsiDATypeTimeStamp [▶ 138] | Datenattribut. |
| Test | FB_AcsiBATypeBOOLEAN [▶ 98] | Datenattribut. |

6.3.1.15.2 FB_AcsiCDC_Cancel_BOOLEAN

Verwendung: SPC, DPC.

Namensraum: Tc3_Acsi [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

FB_AcsiCommonNodeClass [▶ 154] -> FB_AcsiCommonDataClass [▶ 148] -> FB_AcsiCDC_Cancel_BOOLEAN

```
FUNCTION_BLOCK FB_AcsiCDC_Cancel_BOOLEAN EXTENDS FB_AcsiCommonDataClass
```

Eingänge

| Name | Typ | Beschreibung |
|--------|-----------------------------------|---|
| ctlVal | FB_AcsiBTypeBOOLEAN
[▶ 98] | Datenattribut: Steuerwert (FALSE:=Off, TRUE:=On). |
| operTm | FB_AcsiDTypeTimeStamp
[▶ 138] | Datenattribut. |
| origin | FB_AcsiDTypeOriginator
[▶ 125] | Datenattribut. |
| ctlNum | FB_AcsiBTypeINT8U [▶ 100] | Datenattribut. |
| T | FB_AcsiDTypeTimeStamp
[▶ 138] | Datenattribut. |
| Test | FB_AcsiBTypeBOOLEAN
[▶ 98] | Datenattribut. |

6.3.1.15.3 FB_AcsiCDC_Cancel_CODEDENUM

Verwendung: BSC, BAC.

Namensraum: Tc3_Acsi [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

FB_AcsiCommonNodeClass [▶ 154] -> FB_AcsiCommonDataClass [▶ 148] -
> FB_AcsiCDC_Cancel_CODEDENUM

```
FUNCTION_BLOCK FB_AcsiCDC_Cancel_CODEDENUM EXTENDS FB_AcsiCommonDataClass
```

Eingänge

| Name | Typ | Beschreibung |
|--------|--------------------------------|----------------|
| ctlVal | FB_AcsiDTypeTcmd [▶ 138] | Datenattribut. |
| operTm | FB_AcsiDTypeTimeStamp [▶ 138] | Datenattribut. |
| origin | FB_AcsiDTypeOriginator [▶ 125] | Datenattribut. |
| ctlNum | FB_AcsiBTypeINT8U [▶ 100] | Datenattribut. |
| T | FB_AcsiDTypeTimeStamp [▶ 138] | Datenattribut. |
| Test | FB_AcsiBTypeBOOLEAN [▶ 98] | Datenattribut. |

6.3.1.15.4 FB_AcsiCDC_Cancel_ENUMERATED

Verwendung: ENC.

Namensraum: Tc3_Acsi [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

FB_AcsiCommonNodeClass [▶ 154] -> FB_AcsiCommonDataClass [▶ 148] -
> FB_AcsiCDC_Cancel_ENUMERATED

```
FUNCTION_BLOCK FB_AcsiCDC_Cancel_ENUMERATED EXTENDS FB_AcsiCommonDataClass
```


 Eingänge

| Name | Typ | Beschreibung |
|--------|--|----------------|
| ctlVal | FB_AcsiCDC_ENUMERATED_ctlVal [▶ 158] | Datenattribut. |
| operTm | FB_AcsiDATypeTimeStamp [▶ 138] | Datenattribut. |
| origin | FB_AcsiDATypeOriginator [▶ 125] | Datenattribut. |
| ctlNum | FB_AcsiBATypeINT8U [▶ 100] | Datenattribut. |
| T | FB_AcsiDATypeTimeStamp [▶ 138] | Datenattribut. |
| Test | FB_AcsiBATypeBOOLEAN [▶ 98] | Datenattribut. |

6.3.1.15.5 FB_AcsiCDC_Cancel_INT32

Verwendung: INC.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCDC_Cancel_INT32](#)

```
FUNCTION_BLOCK FB_AcsiCDC_Cancel_INT32 EXTENDS FB_AcsiCommonDataClass
```

 Eingänge

| Name | Typ | Beschreibung |
|--------|---|----------------|
| ctlVal | FB_AcsiBATypeINT32 [▶ 101] | Datenattribut. |
| operTm | FB_AcsiDATypeTimeStamp [▶ 138] | Datenattribut. |
| origin | FB_AcsiDATypeOriginator [▶ 125] | Datenattribut. |
| ctlNum | FB_AcsiBATypeINT8U [▶ 100] | Datenattribut. |
| T | FB_AcsiDATypeTimeStamp [▶ 138] | Datenattribut. |
| Test | FB_AcsiBATypeBOOLEAN [▶ 98] | Datenattribut. |

6.3.1.15.6 FB_AcsiCDC_Cancel_INT8

Verwendung: ISC.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -> [FB_AcsiCDC_Cancel_INT8](#)

```
FUNCTION_BLOCK FB_AcsiCDC_Cancel_INT8 EXTENDS FB_AcsiCommonDataClass
```

Eingänge

| Name | Typ | Beschreibung |
|--------|--|----------------|
| ctlVal | FB_AcsiBTypeINT8 [▶ 99] | Datenattribut. |
| operTm | FB_AcsiDTypeTimeStamp [▶ 138] | Datenattribut. |
| Origin | FB_AcsiDTypeOriginator [▶ 125] | Datenattribut. |
| ctlNum | FB_AcsiBTypeINT8U [▶ 100] | Datenattribut. |
| T | FB_AcsiDTypeTimeStamp [▶ 138] | Datenattribut. |
| Test | FB_AcsiBTypeBOOLEAN [▶ 98] | Datenattribut. |

6.3.1.15.7 FB_AcsiCDC_ENUMERATED_ctlVal

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeENUMERATED \[▶ 98\]](#) -> [FB_AcsiCDC_ENUMERATED_ctlVal](#)

```
FUNCTION_BLOCK FB_AcsiCDC_ENUMERATED_ctlVal EXTENDS FB_AcsiBTypeENUMERATED
```

Eigenschaften

-

6.3.1.15.8 FB_AcsiCDC_Oper_AnalogueValue

Verwendung: APC.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCDC_Oper_AnalogueValue](#)

```
FUNCTION_BLOCK FB_AcsiCDC_Oper_AnalogueValue EXTENDS FB_AcsiCommonDataClass
```

Eingänge

| Name | Typ | Beschreibung |
|--------|---|----------------|
| ctlVal | FB_AcsiDTypeAnalogueValue [▶ 105] | Datenattribut. |
| operTm | FB_AcsiDTypeTimeStamp [▶ 138] | Datenattribut. |
| origin | FB_AcsiDTypeOriginator [▶ 125] | Datenattribut. |
| ctlNum | FB_AcsiBTypeINT8U [▶ 100] | Datenattribut. |
| T | FB_AcsiDTypeTimeStamp [▶ 138] | Datenattribut. |
| Test | FB_AcsiBTypeBOOLEAN [▶ 98] | Datenattribut. |
| Check | FB_AcsiDTypeCheck [▶ 109] | Datenattribut. |

6.3.1.15.9 FB_AcsiCDC_Oper_BOOLEAN

Verwendung: SPC, DPC.

Namensraum: [Tc3_Acsi \[▸ 96\]](#)
Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▸ 154\]](#) -> [FB_AcsiCommonDataClass \[▸ 148\]](#) -> [FB_AcsiCDC_Oper_BOOLEAN](#)

```
FUNCTION_BLOCK FB_AcsiCDC_Oper_BOOLEAN EXTENDS FB_AcsiCommonDataClass
```

 **Eingänge**

| Name | Typ | Beschreibung |
|--------|--|---|
| ctlVal | FB_AcsiBTypeBOOLEAN [▸ 98] | Datenattribut: Steuerwert (FALSE:=Off, TRUE:=On). |
| operTm | FB_AcsiDTypeTimeStamp [▸ 138] | Datenattribut. |
| origin | FB_AcsiDTypeOriginator [▸ 125] | Datenattribut. |
| ctlNum | FB_AcsiBTypeINT8U [▸ 100] | Datenattribut. |
| T | FB_AcsiDTypeTimeStamp [▸ 138] | Datenattribut. |
| Test | FB_AcsiBTypeBOOLEAN [▸ 98] | Datenattribut. |
| Check | FB_AcsiDTypeCheck [▸ 109] | Datenattribut. |

6.3.1.15.10 FB_AcsiCDC_Oper_CODEDENUM

Verwendung: BSC, BAC.

Namensraum: [Tc3_Acsi \[▸ 96\]](#)
Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▸ 154\]](#) -> [FB_AcsiCommonDataClass \[▸ 148\]](#) -> [FB_AcsiCDC_Oper_CODEDENUM](#)

```
FUNCTION_BLOCK FB_AcsiCDC_Oper_CODEDENUM EXTENDS FB_AcsiCommonDataClass
```

 **Eingänge**

| Name | Typ | Beschreibung |
|--------|--|----------------|
| ctlVal | FB_AcsiDTypeTcmd [▸ 138] | Datenattribut. |
| operTm | FB_AcsiDTypeTimeStamp [▸ 138] | Datenattribut. |
| origin | FB_AcsiDTypeOriginator [▸ 125] | Datenattribut. |
| ctlNum | FB_AcsiBTypeINT8U [▸ 100] | Datenattribut. |
| T | FB_AcsiDTypeTimeStamp [▸ 138] | Datenattribut. |
| Test | FB_AcsiBTypeBOOLEAN [▸ 98] | Datenattribut. |
| Check | FB_AcsiDTypeCheck [▸ 109] | Datenattribut. |

6.3.1.15.11 FB_AcsiCDC_Oper_ENUMERATED

Verwendung: ENC.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -
> [FB_AcsiCDC_Oper_ENUMERATED](#)

```
FUNCTION_BLOCK FB_AcsiCDC_Oper_ENUMERATED EXTENDS FB_AcsiCommonDataClass
```

Eingänge

| Name | Typ | Beschreibung |
|--------|--|----------------|
| ctlVal | FB_AcsiCDC_ENUMERATED_ctlVal [▶ 158] | Datenattribut. |
| operTm | FB_AcsiDATypeTimeStamp [▶ 138] | Datenattribut. |
| origin | FB_AcsiDATypeOriginator [▶ 125] | Datenattribut. |
| ctlNum | FB_AcsiBATypeINT8U [▶ 100] | Datenattribut. |
| T | FB_AcsiDATypeTimeStamp [▶ 138] | Datenattribut. |
| Test | FB_AcsiBATypeBOOLEAN [▶ 98] | Datenattribut. |
| Check | FB_AcsiDATypeCheck [▶ 109] | Datenattribut. |

6.3.1.15.12 [FB_AcsiCDC_Oper_INT32](#)

Verwendung: INC.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCDC_Oper_INT32](#)

```
FUNCTION_BLOCK FB_AcsiCDC_Oper_INT32 EXTENDS FB_AcsiCommonDataClass
```

Eingänge

| Name | Typ | Beschreibung |
|--------|---|----------------|
| ctlVal | FB_AcsiBATypeINT32 [▶ 101] | Datenattribut. |
| operTm | FB_AcsiDATypeTimeStamp [▶ 138] | Datenattribut. |
| origin | FB_AcsiDATypeOriginator [▶ 125] | Datenattribut. |
| ctlNum | FB_AcsiBATypeINT8U [▶ 100] | Datenattribut. |
| T | FB_AcsiDATypeTimeStamp [▶ 138] | Datenattribut. |
| Test | FB_AcsiBATypeBOOLEAN [▶ 98] | Datenattribut. |
| Check | FB_AcsiDATypeCheck [▶ 109] | Datenattribut. |

6.3.1.15.13 [FB_AcsiCDC_Oper_INT8](#)

Verwendung: ISC.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCDC_Oper_INT8](#)

```
FUNCTION_BLOCK FB_AcsiCDC_Oper_INT8 EXTENDS FB_AcsiCommonDataClass
```

 **Eingänge**

| Name | Typ | Beschreibung |
|--------|--|----------------|
| ctlVal | FB_AcsiBTypeINT8 [▶ 99] | Datenattribut. |
| operTm | FB_AcsiDTypeTimeStamp [▶ 138] | Datenattribut. |
| origin | FB_AcsiDTypeOriginator [▶ 125] | Datenattribut. |
| ctlNum | FB_AcsiBTypeINT8U [▶ 100] | Datenattribut. |
| T | FB_AcsiDTypeTimeStamp [▶ 138] | Datenattribut. |
| Test | FB_AcsiBTypeBOOLEAN [▶ 98] | Datenattribut. |
| Check | FB_AcsiDTypeCheck [▶ 109] | Datenattribut. |

6.3.1.15.14 FB_AcsiCDC_SBO

Select before operate.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> [FB_AcsiCommonAttributeClass \[▶ 146\]](#) -> [FB_AcsiBTypeVISIBLESTRING \[▶ 103\]](#) -> **FB_AcsiCDC_SBO**

```
FUNCTION_BLOCK FB_AcsiCDC_SBO EXTENDS FB_AcsiBTypeVISIBLESTRING
```

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------|-------------|---------|---------------|
| sValue | STRING(129) | Get,Set | Attributwert. |

6.3.1.15.15 FB_AcsiCDC_SBOw_AnalogueValue

Verwendung: APC.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> **FB_AcsiCDC_SBOw_AnalogueValue**

```
FUNCTION_BLOCK FB_AcsiCDC_SBOw_AnalogueValue EXTENDS FB_AcsiCommonDataClass
```

 **Eingänge**

| Name | Typ | Beschreibung |
|--------|---|----------------|
| ctlVal | FB_AcsiDTypeAnalogueValue [▶ 105] | Datenattribut. |
| operTm | FB_AcsiDTypeTimeStamp [▶ 138] | Datenattribut. |
| origin | FB_AcsiDTypeOriginator [▶ 125] | Datenattribut. |
| ctlNum | FB_AcsiBTypeINT8U [▶ 100] | Datenattribut. |
| T | FB_AcsiDTypeTimeStamp [▶ 138] | Datenattribut. |
| Test | FB_AcsiBTypeBOOLEAN [▶ 98] | Datenattribut. |
| Check | FB_AcsiDTypeCheck [▶ 109] | Datenattribut. |

6.3.1.15.16 FB_AcsiCDC_SBOw_BOOLEAN

Verwendung: SPC, DPC.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> FB_AcsiCDC_SBOw_BOOLEAN

```
FUNCTION_BLOCK FB_AcsiCDC_SBOw_BOOLEAN EXTENDS FB_AcsiCommonDataClass
```

Eingänge

| Name | Typ | Beschreibung |
|--------|--|---|
| ctlVal | FB_AcsiBTypeBOOLEAN [▶ 98] | Datenattribut: Steuerwert (FALSE:=Off, TRUE:=On). |
| operTm | FB_AcsiDTypeTimeStamp [▶ 138] | Datenattribut |
| origin | FB_AcsiDTypeOriginator [▶ 125] | Datenattribut |
| ctlNum | FB_AcsiBTypeINT8U [▶ 100] | Datenattribut |
| T | FB_AcsiDTypeTimeStamp [▶ 138] | Datenattribut |
| Test | FB_AcsiBTypeBOOLEAN [▶ 98] | Datenattribut |
| Check | FB_AcsiDTypeCheck [▶ 109] | Datenattribut |

6.3.1.15.17 FB_AcsiCDC_SBOw_CODEDENUM

Verwendung: BSC, BAC.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -> FB_AcsiCDC_SBOw_CODEDENUM

```
FUNCTION_BLOCK FB_AcsiCDC_SBOw_CODEDENUM EXTENDS FB_AcsiCommonDataClass
```

Eingänge

| Name | Typ | Beschreibung |
|--------|--|----------------|
| ctlVal | FB_AcsiDTypeTcmd [▶ 138] | Datenattribut. |
| operTm | FB_AcsiDTypeTimeStamp [▶ 138] | Datenattribut. |
| origin | FB_AcsiDTypeOriginator [▶ 125] | Datenattribut. |
| ctlNum | FB_AcsiBTypeINT8U [▶ 100] | Datenattribut. |
| T | FB_AcsiDTypeTimeStamp [▶ 138] | Datenattribut. |
| Test | FB_AcsiBTypeBOOLEAN [▶ 98] | Datenattribut. |
| Check | FB_AcsiDTypeCheck [▶ 109] | Datenattribut. |

6.3.1.15.18 FB_AcsiCDC_SBOw_ENUMERATED

Verwendung: ENC.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

FB_AcsiCommonNodeClass [▶ 154] -> FB_AcsiCommonDataClass [▶ 148] -
 > FB_AcsiCDC_SBOw_ENUMERATED

```
FUNCTION_BLOCK FB_AcsiCDC_SBOw_ENUMERATED EXTENDS FB_AcsiCommonDataClass
```

 **Eingänge**

| Name | Typ | Beschreibung |
|--------|--------------------------------------|----------------|
| ctlVal | FB_AcsiCDC_ENUMERATED_ctlVal [▶ 158] | Datenattribut. |
| operTm | FB_AcsiDATypeTimeStamp [▶ 138] | Datenattribut. |
| origin | FB_AcsiDATypeOriginator [▶ 125] | Datenattribut. |
| ctlNum | FB_AcsiBATypeINT8U [▶ 100] | Datenattribut. |
| T | FB_AcsiDATypeTimeStamp [▶ 138] | Datenattribut. |
| Test | FB_AcsiBATypeBOOLEAN [▶ 98] | Datenattribut. |
| Check | FB_AcsiDATypeCheck [▶ 109] | Datenattribut. |

6.3.1.15.19 FB_AcsiCDC_SBOw_INT32

Verwendung: INC.

Namensraum: Tc3_Acsi [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

FB_AcsiCommonNodeClass [▶ 154] -> FB_AcsiCommonDataClass [▶ 148] -> FB_AcsiCDC_SBOw_INT32

```
FUNCTION_BLOCK FB_AcsiCDC_SBOw_INT32 EXTENDS FB_AcsiCommonDataClass
```

 **Eingänge**

| Name | Typ | Beschreibung |
|--------|---------------------------------|----------------|
| ctlVal | FB_AcsiBATypeINT32 [▶ 101] | Datenattribut. |
| operTm | FB_AcsiDATypeTimeStamp [▶ 138] | Datenattribut. |
| origin | FB_AcsiDATypeOriginator [▶ 125] | Datenattribut. |
| ctlNum | FB_AcsiBATypeINT8U [▶ 100] | Datenattribut. |
| T | FB_AcsiDATypeTimeStamp [▶ 138] | Datenattribut. |
| Test | FB_AcsiBATypeBOOLEAN [▶ 98] | Datenattribut. |
| Check | FB_AcsiDATypeCheck [▶ 109] | Datenattribut. |

6.3.1.15.20 FB_AcsiCDC_SBOw_INT8

Verwendung: ISC.

Namensraum: Tc3_Acsi [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

FB_AcsiCommonNodeClass [▶ 154] -> FB_AcsiCommonDataClass [▶ 148] -> FB_AcsiCDC_SBOw_INT8

```
FUNCTION_BLOCK FB_AcsiCDC_SBOw_INT8 EXTENDS FB_AcsiCommonDataClass
```

Eingänge

| Name | Typ | Beschreibung |
|--------|---|----------------|
| ctlVal | FB_AcsiBATypeINT8 [▶ 99] | Datenattribut. |
| operTm | FB_AcsiDATypeTimeStamp [▶ 138] | Datenattribut. |
| origin | FB_AcsiDATypeOriginator [▶ 125] | Datenattribut. |
| ctlNum | FB_AcsiBATypeINT8U [▶ 100] | Datenattribut. |
| T | FB_AcsiDATypeTimeStamp [▶ 138] | Datenattribut. |
| Test | FB_AcsiBATypeBOOLEAN [▶ 98] | Datenattribut. |
| Check | FB_AcsiDATypeCheck [▶ 109] | Datenattribut. |

6.3.1.16 Steuerblöcke

6.3.1.16.1 FB_AcsiBufferedReportControlBlock

Der Funktionsbaustein FB_AcsiBufferedReportControlBlock beinhaltet die Standard-Datenattribute eines Berichtssteuerblocks für gepufferte Reports.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -

> [FB_AcsiCommonControlBlockClass \[▶ 147\]](#) -> [FB_AcsiCommonBufferedReportControlBlockClass \[▶ 147\]](#) -

> [FB_AcsiBufferedReportControlBlock](#)

```
FUNCTION_BLOCK FB_AcsiBufferedReportControlBlock EXTENDS FB_AcsiCommonBufferedReportControlBlockClass
```

Eingänge

| Name | Typ | Beschreibung |
|-------------|--|---------------------------------------|
| RptID | FB_AcsiDATypeVisString129 [▶ 144] | Datenattribut, Ed1=>RptID:VisStr[65]! |
| RptEna | FB_AcsiBATypeBOOLEAN [▶ 98] | Datenattribut. |
| DatSet | FB_AcsiDATypeDatSetRef [▶ 146] | Datenattribut. |
| ConfRev | FB_AcsiBATypeINT32U [▶ 101] | Datenattribut. |
| OptFlds | FB_AcsiDATypeOptFlds [▶ 123] | Datenattribut. |
| BufTm | FB_AcsiBATypeINT32U [▶ 101] | Datenattribut. |
| SqNum | FB_AcsiBATypeINT16U [▶ 100] | Datenattribut. |
| TrgOps | FB_AcsiDATypeTriggerConditions [▶ 141] | Datenattribut. |
| IntgPd | FB_AcsiBATypeINT32U [▶ 101] | Datenattribut. |
| GI | FB_AcsiBATypeBOOLEAN [▶ 98] | Datenattribut. |
| PurgeBuf | FB_AcsiBATypeBOOLEAN [▶ 98] | Datenattribut. |
| EntryID | FB_AcsiDATypeEntryID [▶ 114] | Datenattribut. |
| TimeOfEntry | FB_AcsiDATypeEntryTime [▶ 114] | Datenattribut. |
| ResvTms | FB_AcsiBATypeINT16 [▶ 100] | Datenattribut, Ed2. |
| Owner | FB_AcsiDATypeOctet64 [▶ 122] | Datenattribut, Ed2. |

6.3.1.16.2 FB_AcsiGooseControlBlock

Der Funktionsbaustein FB_AcsiGooseControlBlock beinhaltet die Standard-Datenattribute eines GOOSE-Steuerblocks.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -
 > [FB_AcsiCommonControlBlockClass \[▶ 147\]](#) -> [FB_AcsiCommonGooseControlBlockClass \[▶ 150\]](#) -
 > [FB_AcsiGooseControlBlock](#)

```
FUNCTION_BLOCK FB_AcsiGooseControlBlock EXTENDS FB_AcsiCommonGooseControlBlockClass
```

Eingänge

| Name | Typ | Beschreibung |
|------------|---|---|
| GoEna | FB_AcsiBATypeBOOLEAN [▶ 98] | Datenattribut, aktiviert/deaktiviert den GOOSE-Steuerblock. |
| GoID | FB_AcsiDATypeVisString129 [▶ 144] | Datenattribut, die für die GOOSE-Nachrichten genutzte ID. |
| DatSet | FB_AcsiDATypeDatSetRef [▶ 146] | Datenattribut, das für die GOOSE-Nachrichten. |
| ConfRev | FB_AcsiBATypeINT32U [▶ 101] | Datenattribut, die Revisionsnummer der Konfiguration des GOOSE-Steuerblocks. |
| NdsCom | FB_AcsiBATypeBOOLEAN [▶ 98] | Datenattribut, zeigt an, ob der GOOSE-Steuerblock noch nicht vollständig konfiguriert ist. |
| DstAddress | FB_AcsiDATypePhyComAddr [▶ 127] | Datenattribut, die Ziel-Adresse der GOOSE-Nachrichten. |
| MinTime | FB_AcsiBATypeINT32U [▶ 101] | Datenattribut, Zeit zwischen dem direkten Senden nach Datenänderung und der ersten Wiederholung in Millisekunden. |
| MaxTime | FB_AcsiBATypeINT32U [▶ 101] | Datenattribut, maximale Zeit zwischen den wiederholten Übertragungen einer GOOSE-Nachricht in Millisekunden. |
| FixedOffs | FB_AcsiBATypeBOOLEAN [▶ 98] | Datenattribut, aktiviert/deaktiviert die Kodierung mit festen anstatt dynamischen Längen. |

6.3.1.16.3 FB_AcsiUnbufferedReportControlBlock

Der Funktionsbaustein FB_AcsiUnbufferedReportControlBlock beinhaltet die Standard-Datenattribute eines Berichtssteuerblocks für nicht gepufferte Reports.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -
 > [FB_AcsiCommonControlBlockClass \[▶ 147\]](#) -> [FB_AcsiCommonUnbufferedReportControlBlockClass \[▶ 154\]](#) -
 > [FB_AcsiUnbufferedReportControlBlock](#)

```
FUNCTION_BLOCK FB_AcsiUnbufferedReportControlBlock EXTENDS FB_AcsiCommonUnbufferedReportControlBlockClass
```

Eingänge

| Name | Typ | Beschreibung |
|---------|--|---------------------------------------|
| RptID | FB_AcsiDATypeVisString129 [▶ 144] | Datenattribut, Ed1=>RptID:VisStr[65]! |
| RptEna | FB_AcsiBTypeBOOLEAN [▶ 98] | Datenattribut |
| Resv | FB_AcsiBTypeBOOLEAN [▶ 98] | Datenattribut |
| DatSet | FB_AcsiDATypeDatSetRef [▶ 146] | Datenattribut |
| ConfRev | FB_AcsiBTypeINT32U [▶ 101] | Datenattribut |
| OptFlds | FB_AcsiDATypeOptFlds [▶ 123] | Datenattribut |
| BufTm | FB_AcsiBTypeINT32U [▶ 101] | Datenattribut |
| SqNum | FB_AcsiBTypeINT8U [▶ 100] | Datenattribut |
| TrgOps | FB_AcsiDATypeTriggerConditions [▶ 141] | Datenattribut |
| IntgPd | FB_AcsiBTypeINT32U [▶ 101] | Datenattribut |
| GI | FB_AcsiBTypeBOOLEAN [▶ 98] | Datenattribut |
| Owner | FB_AcsiDATypeOctet64 [▶ 122] | Datenattribut, Ed2. |

6.3.2 Protokolleinstellungen

6.3.2.1 FB_GseAdapterSettingsClass

Der Funktionsbaustein FB_GseAdapterSettingsClass dient der Konfiguration eines für die GSE-Kommunikation nutzbaren Netzwerkadapters.

Namensraum: [Tc3_Gse](#) [[▶ 97](#)]

Bibliothek: Tc3_Gse (Tc3_Gse.compiled-library)

Vererbungshierarchie

FB_GseAdapterSettingsClass

```
FUNCTION_BLOCK FB_GseAdapterSettingsClass IMPLEMENTS I_GseAdapterSettingsClass
```

Schnittstellen

| Typ | Beschreibung |
|---|---|
| I_GseAdapterSettingsClass [▶ 322] | Schnittstelle zur Konfiguration des Netzwerkadapters. |

 **Eigenschaften**

| Name | Typ | Zugriff | Definitionsort | Initialwert | Beschreibung |
|------------------|--------------------------------------|---------|--|--|--|
| bExecGoCBPubs | BOOL | Get | I_GseAdapterSettingsClass
[► 322] | FALSE | Aktiviert/Deaktiviert das automatische Aufrufen der „Execute“-Methode der Publisher GOOSE-Kontrollblöcke durch den Adapter. |
| | | Set | Lokal | | |
| bExecGoCBSubs | BOOL | Get | I_GseAdapterSettingsClass
[► 322] | FALSE | Aktiviert/Deaktiviert das automatische Aufrufen der „Execute“-Methode der Subscriber GOOSE-Kontrollblöcke durch den Adapter. |
| | | Set | Lokal | | |
| bSimulation | BOOL | Get | I_GseAdapterSettingsClass
[► 322] | FALSE | Aktiviert/Deaktiviert das GOOSE-Simulations-Bit Feld. |
| | | Set | Lokal | | |
| eAdapterPriority | TcEthernetAdapterPriority
[► 453] | Get | I_GseAdapterSettingsClass
[► 322] | TcEthernetAdapterPriority.High | Priorität des Netzwerkadapters. |
| | | Set | Lokal | | |
| eDispatchMode | E_GseDispatchMode
[► 445] | Get | I_GseAdapterSettingsClass
[► 322] | E_GseDispatchMode.
NonPromiscuous | Einstellung, ob empfangene Ethernet-Telegramme an das Betriebssystem weitergeleitet werden. |
| | | Set | Lokal | | |
| macAddr | ETHERNET ADDRESS
[► 451] | Get | I_GseAdapterSettingsClass
[► 322] | [0, 0, 0, 0, 0, 0]
(nicht gesetzt, '00-00-00-00-00-00') | MAC-Adresse des Netzwerkadapters. |
| | | Set | I_GseAdapterSettingsClass
[► 322] | | |

| Name | Typ | Zugriff | Definitionsort | Initialwert | Beschreibung |
|-------------------|---|---------|--|--|--|
| multicast
Addr | ETHERNET_ADDRESS
[▶ 451] | Get | I_GseAdapterSettingsClass
[▶ 322] | [0, 0, 0, 0, 0, 0]
(nicht gesetzt, '00-00-00-00-00-00') | Ausgewählte, einzelne Subscriber Multicast-Adresse des Netzwerkadapters. Wenn gesetzt, empfängt die Netzwerkadapterbausteininstanz nur die Subscriber-Nachrichten, die exakt an diese Adresse gesendet werden. In diesem Fall wird die Adresse als Filter für die Subscriber-Nachrichten verwendet. Alle anderen Nachrichten werden entweder an das Betriebssystem oder (wenn vorhanden) an weitere Instanzen des Netzwerkadapterbausteins weitergeleitet.

Hinweis Ab der TwinCAT TF6510 IEC 61850 Telecontrol v3.1.98.0 und höher: Wenn diese Adresse nicht gesetzt wurde ('00-00-00-00-00-00'), dann werden als Filter für die Subscriber-Nachrichten alle Multicast-Adressen und APPIDs verwendet, die an den Subscriber Steuerblockinstanzen konfiguriert wurden. In diesem Fall können z.B. in den GOOSE-Steuerblockinstanzen unterschiedliche Multicast-Adressen konfiguriert werden. Die konfigurierten Subscriber APPIDs (wenn verwendet > 0), müssen den APPIDs auf der Publisher-Seite entsprechen. Alle empfangenen Nachrichten, die keinem Steuerblock zugeordnet werden können, werden an das Betriebssystem oder an weitere Instanzen des Netzwerkadapterbausteins weitergeleitet. |
| | | Set | Lokal | | |
| oid | OTCID
[▶ 452] | Get | I_GseAdapterSettingsClass
[▶ 322] | 0 | TwinCAT-Object-ID des Netzwerkadapters. Zur Konfiguration des Parameters siehe: RT - Ethernet Adapter Konfiguration [▶ 37]. |
| | | Set | Lokal | | |
| sMacAddr | STRING(17) | Get | I_GseAdapterSettingsClass
[▶ 322] | '00-00-00-00-00-00' (nicht gesetzt) | Textrepräsentation der MAC-Adresse des Netzwerkadapters. |
| sMulticastAddr | STRING(17) | Get | I_GseAdapterSettingsClass
[▶ 322] | '00-00-00-00-00-00' (nicht gesetzt) | Textrepräsentation der Multicast-Adresse des Netzwerkadapters. |
| | | Set | I_GseAdapterSettingsClass
[▶ 322] | | |

6.3.2.2 FB_MmsClientSettingsClass

Der Funktionsbaustein FB_MmsClientSettingsClass dient der Konfiguration der MMS-Kommunikationsschicht.

Namensraum: [Tc3_Mms](#) [▶ 97]

Bibliothek: Tc3_Mms (Tc3_Mms.compiled-library)

Vererbungshierarchie

FB SocketClientSettingsClass [▶ 200] -> FB TpktClientSettingsClass [▶ 205] -> FB Rfc1006ClientSettingsClass [▶ 183] -> FB UlosiClientSettingsClass [▶ 210] -> FB_MmsClientSettingsClass

FUNCTION_BLOCK FB_MmsClientSettingsClass EXTENDS FB_UlosiClientSettingsClass IMPLEMENTS I_MmsClientSettingsClass

Schnittstellen

| Typ | Beschreibung |
|----------------------------------|--|
| I_MmsClientSettingsClass [▶ 323] | Schnittstelle zur Konfiguration der MMS-Kommunikationsschicht. |

Eingänge

| Name | Typ | Initialwert | Beschreibung |
|--------------------|--|--|---|
| services | FB_MmsServiceSupportOptions [▶ 175] | .nUsedBits :=
cMmsServicesSupportOptionsBitSize (* default: 85 *),
.bStatus := TRUE,
.bGetNameList := TRUE,
.bIdentify := TRUE,
.bRead := TRUE,
.bWrite := TRUE,
.bGetVariableAccessAttributes := TRUE,
.bDefineNamedVariableList := TRUE,
.bGetNamedVariableListAttributes := TRUE,
.bDeleteNamedVariableList := TRUE,
.bGetDomainAttributes := TRUE,
.bInformationReport := TRUE,
.bConclude := TRUE,
.bCancel := TRUE | Objekt zur Konfiguration der unterstützten MMS-Protokolldienste. |
| parameters | FB_MmsParameterSupportOptions [▶ 175] | .nUsedBits := .cMmsParameterSupportOptionsBitSize (* default := 11 *),
.bStr1 := TRUE,
.bStr2 := TRUE,
.bVnam := TRUE,
.bValt := TRUE,
.bVlis := TRUE | Objekt zur Konfiguration der unterstützten MMS-Protokollparameter. |
| extendedServices | FB_MmsAdditionalSupportOptions [▶ 176] | .nUsedBits :=
cMmsAdditionalSupportOptionsBitSize (* default := 0, not used *) | Objekt zur Konfiguration der zusätzlich unterstützten MMS-Protokolldienste. |
| extendedParameters | FB_MmsAdditionalCBBOptions [▶ 176] | .nUsedBits :=
cMmsAdditionalCbbOptionsBitSize (* default := 0, not used *) | Objekt zur Konfiguration der zusätzlich unterstützten MMS-CBB-Protokollparameter. |

 **Eigenschaften**

| Name | Typ | Zugriff | Definitionsart | Initialwert | Beschreibung |
|---------------------|--|---------|--|---|--|
| bAutoCleanup | BOOL | Get,Set | Geerbt von FB_SocketClientSettingsClass [▶ 200] | TRUE | Aktiviert/Deaktiviert das automatische Aufräumen/Schließen der unbenutzten Sockets (z.B. nach Programmdownload oder Reset der SPS). |
| sSrvNetID | T_AmsNetID [▶ 452] | Get,Set | -/- | “ | TwinCAT Netzwerkadresse des TwinCAT TCP/IP Servers (TF6310). Beim Leerstring wird die Netzwerkadresse des lokalen TwinCAT Systems verwendet. |
| nRemotePort | UDINT | Get,Set | -/- | ‘127.0.0.1’ | TCP/IP Portnummer des Servers. |
| sRemoteHost | T_Ipv4Addr [▶ 452] | Get,Set | -/- | 102 | IPv4-Netzwerkadresse des Servers. |
| tConnect | TIME | Get,Set | -/- | T#45s | Früheste Zeit, nach deren Ablauf eine neue Verbindung aufgebaut werden kann. |
| tAdsTimeout | TIME | Get,Set | -/- | DEFAULT_ADS_TIMEOUT | Maximale Timeoutzeit für ADS-Dienste, sie darf nicht überschritten werden. |
| tSocketTimeout | TIME | Get,Set | -/- | T#5s | Maximale Timeoutzeit die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| bThrottleMode | BOOL | Get,Set | -/- | TRUE | Aktiviert/deaktiviert den Empfangsdaten-Throttle-Polling-Mode. |
| throttleTimes | T_ThrottleTimes [▶ 453] | Get,Set | -/- | Param_Sockets.cDefaultReceiverThrottleTimes | Konfiguration der Zykluszeiten für den Empfangsdaten-Throttle-Polling-Mode. |
| eTraceSocket | E_TraceLevel [▶ 449] | Get,Set | -/- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| ipTLS | I_SocketTlsSettingsClass [▶ 331] | Get,Set | -/- | 0 | Konfigurationsparameter der TLS-Protokollsicht (Transport Layer Security). Wert Null = nicht konfiguriert. |
| tTpktTimeout | TIME | Get,Set | Geerbt von FB_TpktClientSettingsClass [▶ 205] | T#10s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceTpkt | E_TraceLevel [▶ 449] | Get,Set | -/- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| sCalling_T_Selector | STRING | Get,Set | Geerbt von FB_Rfc1006ClientSettingsClass [▶ 183] | ‘00 00’ | RFC 1006 Parameter: Calling transport address selector (source). |

| Name | Typ | Zugriff | Definitionsort | Initialwert | Beschreibung |
|-----------------------|---|---------|--|-----------------------------|--|
| sCalled_T_Selector | STRING | Get,Set | --- | '00 01' | RFC 1006 Parameter: Called transport address selector (destination). |
| eTpduSize | E_Rfc1006TpduSize [▶ 448] | Get,Set | --- | E_Rfc1006TpduSize.Size_1024 | Maximale TPDU-Bytelänge. |
| tRfc1006Timeout | TIME | Get,Set | --- | T#15s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceRfc1006 | E_TraceLevel [▶ 449] | Get,Set | --- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| sCalling_S_Selector | STRING | Get,Set | Geerbt von FB_UlosiClientSettingsClass [▶ 210] | '00 01' | Calling-S-selector. |
| sCalled_S_Selector | STRING | Get,Set | --- | '00 01' | Called-S-selector. |
| nSessionRequirement | WORD | Get,Set | --- | 2 | Session requirements. |
| sCalling_P_Selector | STRING | Get,Set | --- | '00 00 00 01' | Calling-P-selector. |
| sCalled_P_Selector | STRING | Get,Set | --- | '00 00 00 01' | Called-P-selector. |
| nProtocolVersion | BYTE | Get,Set | --- | 1 | Protocol version. |
| sContext_Name | STRING | Get,Set | --- | '1.0.9506.2.3' | Application context name (object identifier, dotted presentation), MMS. |
| sCalled_AP_Title | STRING | Get,Set | --- | '1.1.1.999.1' | Called-AP-title. |
| nCalled_AE_Qualifier | UDINT | Get,Set | --- | 12 | Called-AE-qualifier. |
| nCalled_AP_InvID | UDINT | Get,Set | --- | 0 | Called-AP-invocation-identifier. |
| nCalled_AE_InvID | UDINT | Get,Set | --- | 0 | Called-AE-invocation-identifier. |
| sCalling_AP_Title | STRING | Get,Set | --- | '1.1.1.999' | Calling-AP-title. |
| nCalling_AE_Qualifier | UDINT | Get,Set | --- | 12 | Calling-AE-qualifier. |
| nCalling_AP_InvID | UDINT | Get,Set | --- | 0 | Calling-AP-invocation-identifier. |
| nCalling_AE_InvID | UDINT | Get,Set | --- | 0 | Calling-AE-invocation-identifier. |
| nRequirements | BYTE | Get,Set | --- | 0 | Requirements (1 == authentication). |
| sMechanism_Name | STRING | Get,Set | --- | '2.2.3.1' | Authentication mechanism name. |
| sAuthent_Value | STRING | Get,Set | --- | '' | Authentication value (password). |

| Name | Typ | Zugriff | Definitionsart | Initialwert | Beschreibung |
|-------------------------|---------------------------------------|------------|---|-------------------|---|
| tUlosiTimeout | TIME | Get, Set | -/- | T#20s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceUlosi | E_TraceLevel [▶ 449] | Get, Set | -/- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| nLocalDetail | T_MmsInteger32 [▶ 478] | Get
Set | I_MmsClientSettingsClass [▶ 323]
Local | 65000 | MMS Initiate-RequestPDU-Parameter: localDetailCalling. |
| nMaxServOutCalling | T_MmsInteger16 [▶ 477] | Get
Set | I_MmsClientSettingsClass [▶ 323]
Local | 10 | MMS Initiate-RequestPDU-Parameter: proposedMaxServOutstanding Calling |
| nMaxServOutCalled | T_MmsInteger16 [▶ 477] | Get
Set | I_MmsClientSettingsClass [▶ 323]
Local | 10 | MMS Initiate-RequestPDU-Parameter: proposedMaxServOutstanding Called. |
| nNestingLevel | T_MmsInteger8 [▶ 477] | Get
Set | I_MmsClientSettingsClass [▶ 323]
Local | 5 | MMS Initiate-RequestPDU-Parameter: proposedDataStructureNesting Level. |
| nMmsVersionNumber | T_MmsInteger16 [▶ 477] | Get
Set | I_MmsClientSettingsClass [▶ 323]
Local | 16#0001 | MMS initRequestDetail-Parameter: proposedVersionNumber. |
| ipParameters | I_MmsParameterSupportOptions [▶ 325] | Get
Set | I_MmsClientSettingsClass [▶ 323]
Local | - | Schnittstellenzeiger vom Objekt zur Konfiguration der MMS initRequestDetail-Parameter: proposedParameterCBB. |
| ipServices | I_MmsServiceSupportOptions [▶ 324] | Get
Set | I_MmsClientSettingsClass [▶ 323]
Local | - | Schnittstellenzeiger vom Objekt zur Konfiguration der MMS initRequestDetail-Parameter: servicesSupportedCalling. |
| ipExtendedServices | I_MmsAdditionalSupportOptions [▶ 325] | Get
Set | I_MmsClientSettingsClass [▶ 323]
Local | - | Schnittstellenzeiger vom Objekt zur Konfiguration der MMS initRequestDetail-Parameter: additionalSupportedCalling. |
| ipExtendedParameters | I_MmsAdditionalCBBOptions [▶ 326] | Get
Set | I_MmsClientSettingsClass [▶ 323]
Local | - | Schnittstellenzeiger vom Objekt zur Konfiguration der MMS initRequestDetail-Parameter: additionalCbbSupportedCalling. |
| sPrivilegeClassIdentity | T_MmsVisibleString [▶ 478] | Get
Set | I_MmsClientSettingsClass [▶ 323]
Local | “ | MMS-initRequestDetail-Parameter: privilegeClassIdentityCalled. |
| tMmsTimeout | TIME | Get
Set | I_MmsClientSettingsClass [▶ 323]
Local | T#25s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceMms | E_TraceLevel [▶ 449] | Get
Set | I_MmsClientSettingsClass [▶ 323]
Local | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |

6.3.2.2.1 FB_MmsServiceSupportOptions

Der Funktionsbaustein FB_MmsServiceSupportOptions dient der Konfiguration der unterstützten MMS-Protokolldienste.

Vererbungshierarchie

FB_MmsServiceSupportOptions

FUNCTION_BLOCK FB_MmsServiceSupportOptions IMPLEMENTS I_MmsServiceSupportOptions

Schnittstellen

| Typ | Beschreibung |
|--|---|
| I_MmsServiceSupportOptions ▶ 324 | Schnittstelle zur Konfiguration der unterstützten MMS-Protokolldienste. |

Eigenschaften

| Name | Typ | Zugriff | Initialwert | Beschreibung |
|---------------------------------|------|---------|-------------|--------------|
| bCancel | BOOL | Get,Set | FALSE | |
| bConclude | BOOL | Get,Set | FALSE | |
| bDefineNamedType | BOOL | Get,Set | FALSE | |
| bDefineNamedVariable | BOOL | Get,Set | FALSE | |
| bDefineNamedVariableList | BOOL | Get,Set | FALSE | |
| bDefineScatteredAccess | BOOL | Get,Set | FALSE | |
| bDeleteNamedType | BOOL | Get,Set | FALSE | |
| bDeleteNamedVariableList | BOOL | Get,Set | FALSE | |
| bDeleteVariableAccess | BOOL | Get,Set | FALSE | |
| bGetDomainAttributes | BOOL | Get,Set | FALSE | |
| bGetNamedTypeAttributes | BOOL | Get,Set | FALSE | |
| bGetNamedVariableListAttributes | BOOL | Get,Set | FALSE | |
| bGetNameList | BOOL | Get,Set | FALSE | |
| bGetScatteredAccessAttributes | BOOL | Get,Set | FALSE | |
| bGetVariableAccessAttributes | BOOL | Get,Set | FALSE | |
| bIdentify | BOOL | Get,Set | FALSE | |
| bInformationReport | BOOL | Get,Set | FALSE | |
| bRead | BOOL | Get,Set | FALSE | |
| bRename | BOOL | Get,Set | FALSE | |
| bStatus | BOOL | Get,Set | FALSE | |
| bWrite | BOOL | Get,Set | FALSE | |

6.3.2.2.2 FB_MmsParameterSupportOptions

Der Funktionsbaustein FB_MmsParameterSupportOptions dient der Konfiguration der unterstützten MMS-Protokollparameter.

Vererbungshierarchie

FB_MmsParameterSupportOptions

FUNCTION_BLOCK FB_MmsParameterSupportOptions IMPLEMENTS I_MmsParameterSupportOptions

Schnittstellen

| Typ | Beschreibung |
|--|---|
| I_MmsParameterSupportOptions ▶ 325 | Schnittstelle zur Konfiguration der unterstützten MMS-Protokollparameter. |

Eigenschaften

| Name | Typ | Zugriff | Initialwert | Beschreibung |
|-------|------|---------|-------------|--|
| bCspi | BOOL | Get,Set | FALSE | |
| bCsr | BOOL | Get,Set | FALSE | |
| bStr1 | BOOL | Get,Set | FALSE | Array-Support |
| bStr2 | BOOL | Get,Set | FALSE | Struktur-Supprt |
| bVadr | BOOL | Get,Set | FALSE | |
| bValt | BOOL | Get,Set | FALSE | Support für Index und Index-Range-Werte (alternate access) |
| bVlis | BOOL | Get,Set | FALSE | Support für namedVariableList |
| bVnam | BOOL | Get,Set | FALSE | Support für namedVariables |

6.3.2.2.3 FB_MmsAdditionalSupportOptions

Der Funktionsbaustein FB_MmsAdditionalSupportOptions dient der Konfiguration der zusätzlich unterstützten MMS-Protokolldienste.

Vererbungshierarchie

FB_MmsAdditionalSupportOptions

FUNCTION_BLOCK FB_MmsAdditionalSupportOptions IMPLEMENTS I_MmsAdditionalSupportOptions

Schnittstellen

| Typ | Beschreibung |
|---|--|
| I_MmsAdditionalSupportOptions ▶ 325 | Schnittstelle zur Konfiguration der zusätzlich unterstützten MMS-Protokolldienste. |

Eigenschaften

| Name | Typ | Zugriff | Initialwert | Beschreibung |
|-----------|------|----------|-------------|--------------|
| bVmdReset | BOOL | Get,Set | FALSE | |
| bVmdStop | BOOL | Get,Set | FALSE | |
| bSelect | BOOL | Get, Set | FALSE | |

6.3.2.2.4 FB_MmsAdditionalCBBOptions

Der Funktionsbaustein FB_MmsAdditionalCBBOptions dient der Konfiguration der zusätzlich unterstützten MMS-CBB-Protokollparameter.

Vererbungshierarchie

FB_MmsAdditionalCBBOptions

FUNCTION_BLOCK FB_MmsAdditionalCBBOptions IMPLEMENTS I_MmsAdditionalCBBOptions

 **Schnittstellen**

| Typ | Beschreibung |
|---|--|
| I_MmsAdditionalCBBOptions ▶ 326 | Schnittstelle zur Konfiguration der zusätzlich unterstützten MMS-CBB-Protokollparameter. |

 **Eigenschaften**

| Name | Typ | Zugriff | Initialwert | Beschreibung |
|------|------|----------|-------------|--------------|
| bDei | BOOL | Get,Set | FALSE | |
| bDes | BOOL | Get,Set | FALSE | |
| bRed | BOOL | Get, Set | FALSE | |

6.3.2.3 FB_MmsServerSettingsClass

Der Funktionsbaustein FB_MmsServerSettingsClass dient der Konfiguration der MMS-Kommunikationsschicht.

Namensraum: Tc3_Mms [▶ 97](#)

Bibliothek: Tc3_Mms (Tc3_Mms.compiled-library)

Vererbungshierarchie

[FB_SocketServerSettingsClass ▶ 201](#) -> [FB_TpktServerSettingsClass ▶ 207](#) -> [FB_Rfc1006ServerSettingsClass ▶ 186](#) -> [FB_UlosiServerSettingsClass ▶ 214](#) -> **FB_MmsServerSettingsClass**

```
FUNCTION_BLOCK FB_MmsServerSettingsClass EXTENDS FB_UlosiServerSettingsClass IMPLEMENTS I_MmsServerSettingsClass
```

 **Schnittstellen**

| Typ | Beschreibung |
|--|--|
| I_MmsServerSettingsClass ▶ 326 | Schnittstelle zur Konfiguration der MMS-Kommunikationsschicht. |

 Eingänge

| Name | Typ | Initialwert | Beschreibung |
|--------------------|--|--|---|
| services | FB MmsServiceSupportOptions [► 175] | .nUsedBits :=
cMmsServicesSupportOptionsBitSize (* default: 85 *),
.bStatus := TRUE,
.bGetNameList := TRUE,
.bIdentify := TRUE,
.bRead := TRUE,
.bWrite := TRUE,
.bGetVariableAccessAttributes := TRUE,
.bDefineNamedVariableList := TRUE,
.bGetNamedVariableListAttributes := TRUE,
.bDeleteNamedVariableList := TRUE,
.bGetDomainAttributes := TRUE,
.bInformationReport := TRUE,
.bConclude := TRUE,
.bCancel := TRUE | Objekt zur Konfiguration der unterstützten MMS-Protokolldienste. |
| parameters | FB MmsParameterSupportOptions [► 175] | .nUsedBits := .cMmsParameterSupportOptionsBitSize (* default := 11 *),
.bStr1 := TRUE,
.bStr2 := TRUE,
.bVnam := TRUE,
.bValt := TRUE,
.bVlis := TRUE | Objekt zur Konfiguration der unterstützten MMS-Protokollparameter. |
| extendedServices | FB MmsAdditionalSupportOptions [► 176] | .nUsedBits :=
cMmsAdditionalSupportOptionsBitSize (* default := 0, not used *) | Objekt zur Konfiguration der zusätzlich unterstützten MMS-Protokolldienste. |
| extendedParameters | FB MmsAdditionalCBBOptions [► 176] | .nUsedBits :=
cMmsAdditionalCbbOptionsBitSize (* default := 0, not used *) | Objekt zur Konfiguration der zusätzlich unterstützten MMS-CBB-Protokollparameter. |

 **Eigenschaften**

| Name | Typ | Zugriff | Definitionsart | Initialwert | Beschreibung |
|----------------|--|---------|---|---|---|
| bAutoCleanup | BOOL | Get,Set | Geerbt von FB_SocketServerSettingsClass [▶ 201] | TRUE | Aktiviert/Deaktiviert das automatische Aufräumen/Schließen der unbenutzten Sockets (z. B. nach Programmdownload oder Reset der SPS). |
| bEnable | BOOL | Get,Set | -//- | FALSE | Aktiviert/deaktiviert das Herstellen der Serververbindungen. Beim „TRUE“ wird Listen-Befehl aktiviert (horchen und ankommende Verbindungen werden akzeptiert) und beim „FALSE“ abgelehnt/deaktiviert. |
| sSrvNetID | T_AmsNetID [▶ 452] | Get,Set | -//- | “ | TwinCAT Netzwerkadresse des TwinCAT TCP/IP Servers (TF6310). Beim Leerstring wird die Netzwerkadresse des lokalen TwinCAT Systems verwendet. |
| nLocalPort | UDINT | Get,Set | -//- | '127.0.0.1' | TCP/IP Portnummer des Servers. |
| sLocalHost | T_Ipv4Address [▶ 452] | Get,Set | -//- | 102 | IPv4-Netzwerkadresse des Servers. |
| tAccept | TIME | Get,Set | -//- | T#1s | Wiederholungszykluszeit für die internen TCP/IP-Socket Accept-Befehle. |
| tListen | TIME | Get,Set | -//- | T#1s | Wiederholungszykluszeit für den internen TCP/IP-Socket Listen-Befehl. |
| tAdsTimeout | TIME | Get,Set | -//- | DEFAULT_ADS_TIMEOUT | Maximale Timeoutzeit für ADS-Dienste, sie darf nicht überschritten werden. |
| tSocketTimeout | TIME | Get,Set | -//- | T#5s | Maximale Timeoutzeit die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| bThrottleMode | BOOL | Get,Set | -//- | TRUE | Aktiviert/deaktiviert den Empfangsdaten-Throttle-Polling-Mode. |
| throttleTimes | T_ThrottleTimes [▶ 453] | Get,Set | -//- | Param_Sockets.cDefaultReceiverThrottleTimes | Konfiguration der Zykluszeiten für den Empfangsdaten-Throttle-Polling-Mode. |
| eTraceSocket | E_TraceLevel [▶ 449] | Get,Set | -//- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| ipTLS | I_SocketTlsSettingsClass [▶ 331] | Get,Set | -//- | 0 | Konfigurationsparameter der TLS-Protokollschicht (Transport Layer Security). Wert Null = nicht konfiguriert. |
| tTpktTimeout | TIME | Get,Set | Geerbt von FB_TpktServerSettingsClass [▶ 207] | T#10s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceTpkt | E_TraceLevel [▶ 449] | Get,Set | -//- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| | | | | | |

| Name | Typ | Zugriff | Definitionsort | Initialwert | Beschreibung |
|-----------------------|---|---------|--|---|--|
| sCalling_T_Selector | STRING | Get,Set | Geerbt von FB_Rfc1006ServerSettingsClass [▶ 186] | '00 00' | RFC 1006 Parameter: Calling transport address selector (source). |
| sCalled_T_Selector | STRING | Get,Set | -//- | '00 01' | RFC 1006 Parameter: Called transport address selector (destination). |
| eTpduSize | E_Rfc1006TpduSize [▶ 448] | Get,Set | -//- | E_Rfc1006TpduSize.Size_1024 | Maximale TPDU-Bytlänge. |
| tRfc1006Timeout | TIME | Get,Set | -//- | T#15s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceRfc1006 | E_TraceLevel [▶ 449] | Get,Set | -//- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| sCalling_S_Selector | STRING | Get,Set | Geerbt von FB_UlosiServerSettingsClass [▶ 214] | '00 01' | Calling-S-selector. |
| sCalled_S_Selector | STRING | Get,Set | -//- | '00 01' | Called-S-selector. |
| nSessionRequirement | WORD | Get,Set | -//- | 2 | Session requirements. |
| sCalling_P_Selector | STRING | Get,Set | -//- | '00 00 00 01' | Calling-P-selector. |
| sCalled_P_Selector | STRING | Get,Set | -//- | '00 00 00 01' | Called-P-selector. |
| nProtocolVersion | BYTE | Get,Set | -//- | 1 | Protocol version. |
| sContextName | STRING | Get,Set | -//- | '1.0.9506.2.3' | Application context name (object identifier, dotted presentation), MMS. |
| sCalled_AP_Title | STRING | Get,Set | -//- | '1.1.1.999.1' | Called-AP-title. |
| nCalled_AE_Qualifier | UDINT | Get,Set | -//- | 12 | Called-AE-qualifier. |
| nCalled_AP_InvID | UDINT | Get,Set | -//- | 0 | Called-AP-invocation-identifier. |
| nCalled_AE_InvID | UDINT | Get,Set | -//- | 0 | Called-AE-invocation-identifier. |
| sCalling_AP_Title | STRING | Get,Set | -//- | '1.1.1.999' | Calling-AP-title. |
| nCalling_AE_Qualifier | UDINT | Get,Set | -//- | 12 | Calling-AE-qualifier. |
| nCalling_AP_InvID | UDINT | Get,Set | -//- | 0 | Calling-AP-invocation-identifier. |
| nCalling_AE_InvID | UDINT | Get,Set | -//- | 0 | Calling-AE-invocation-identifier. |
| nRequirements | BYTE | Get,Set | -//- | 0 | Requirements (1 == authentication). |
| sMechanismName | STRING | Get,Set | -//- | '2.2.3.1' | Authentication mechanism name. |

| Name | Typ | Zugriff | Definitionsort | Initialwert | Beschreibung |
|--------------------|--|------------|--|-------------------|--|
| sAuthent_Value | STRING | Get,Set | -/- | “ | Authentication value (password). |
| tUloSiTimeOut | TIME | Get,Set | -/- | T#20s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceUloSi | E_TraceLevel
[► 449] | Get,Set | -/- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| nLocalDetail | T_MmsInteger32
[► 478] | Get
Set | I_MmsServerSettingsClass
[► 326]
Local | 65000 | MMS Initiate-RequestPDU-Parameter: localDetailCalling. |
| nMaxServOutCalling | T_MmsInteger16
[► 477] | Get
Set | I_MmsServerSettingsClass
[► 326]
Local | 10 | MMS Initiate-RequestPDU-Parameter: proposedMaxServOutstandingCalling |
| nMaxServOutCalled | T_MmsInteger16
[► 477] | Get
Set | I_MmsServerSettingsClass
[► 326]
Local | 10 | MMS Initiate-RequestPDU-Parameter: proposedMaxServOutstandingCalled. |
| nNestingLevel | T_MmsInteger8
[► 477] | Get
Set | I_MmsServerSettingsClass
[► 326]
Local | 5 | MMS Initiate-RequestPDU-Parameter: proposedDataStructureNestingLevel. |
| nMmsVersionNumber | T_MmsInteger16
[► 477] | Get
Set | I_MmsServerSettingsClass
[► 326]
Local | 16#0001 | MMS initRequestDetail-Parameter: proposedVersionNumber. |
| ipParameters | I_MmsParameterSupportOptions
[► 325] | Get
Set | I_MmsServerSettingsClass
[► 326]
Local | - | Schnittstellenzeiger vom Objekt zur Konfiguration der MMS initRequestDetail-Parameter: proposedParameterCBB. |
| ipServices | I_MmsServiceSupportOptions
[► 324] | Get
Set | I_MmsServerSettingsClass
[► 326]
Local | - | Schnittstellenzeiger vom Objekt zur Konfiguration der MMS initRequestDetail-Parameter: servicesSupportedCalling. |
| ipExtendedServices | I_MmsAdditionalSupportOptions
[► 325] | Get
Set | I_MmsServerSettingsClass
[► 326]
Local | - | Schnittstellenzeiger vom Objekt zur Konfiguration der MMS initRequestDetail-Parameter: additionalSupportedCalling. |

| Name | Typ | Zugriff | Definitionsort | Initialwert | Beschreibung |
|-------------------------|-----------------------------------|---------|----------------------------------|-------------------|---|
| ipExtendedParameter | I_MmsAdditionalCBBOptions [▶ 326] | Get | I_MmsServerSettingsClass [▶ 326] | - | Schnittstellenzeiger vom Objekt zur Konfiguration der MMS initRequestDetail-Parameter: additionalCbbSupportedCalling. |
| | | Set | Local | | |
| sPrivilegeClassIdentity | T_MmsVisibleString [▶ 478] | Get | I_MmsServerSettingsClass [▶ 326] | " | MMS-initRequestDetail-Parameter: privilegeClassIdentityCalled. |
| | | Set | Local | | |
| tMmsTimeout | TIME | Get | I_MmsServerSettingsClass [▶ 326] | T#25s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| | | Set | Local | | |
| eTraceMms | E_TraceLevel [▶ 449] | Get | I_MmsServerSettingsClass [▶ 326] | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| | | Set | Local | | |

6.3.2.4 FB_Rfc1006ClientSettingsClass

Der Funktionsbaustein FB_Rfc1006ClientSettingsClass dient der Konfiguration der RFC1006-Kommunikationsschicht.

Namensraum: Tc3_Rfc1006 [▶ 97]

Bibliothek: Tc3_Rfc1006 (Tc3_Rfc1006.compiled-library)

Vererbungshierarchie

FB_SocketClientSettingsClass [▶ 200] -> FB_TpktClientSettingsClass [▶ 205] -> FB_Rfc1006ClientSettingsClass

```
FUNCTION_BLOCK FB_Rfc1006ClientSettingsClass EXTENDS FB_TpktClientSettingsClass IMPLEMENTS I_Rfc1006ClientSettingsClass
```

Schnittstellen

| Typ | Beschreibung |
|--------------------------------------|---|
| I_Rfc1006ClientSettingsClass [▶ 327] | Schnittstelle zur Konfiguration der RFC1006-Kommunikationsschicht |

 **Eigenschaften**

| Name | Typ | Zugriff | Definitionsort | Initialwert | Beschreibung |
|---------------------|---|------------|---|---|--|
| bAutoClean
up | BOOL | Get, Set | Geerbt von
FB_SocketClientSettingsClass [► 200] | TRUE | Aktiviert/Deaktiviert das automatische Aufräumen/Schließen der unbenutzten Sockets (z.B. nach Programmdownload oder Reset der SPS). |
| sSrvNetID | T_AmsNetID
[► 452] | Get, Set | -/- | “ | TwinCAT Netzwerkadresse des TwinCAT TCP/IP Servers (TF6310). Beim Leerstring wird die Netzwerkadresse des lokalen TwinCAT Systems verwendet. |
| nRemotePort | UDINT | Get, Set | -/- | ‘127.0.0.1’ | TCP/IP Portnummer des Servers. |
| sRemoteHost | T_Ipv4Addr
[► 452] | Get, Set | -/- | 102 | IPv4-Netzwerkadresse des Servers. |
| tConnect | TIME | Get, Set | -/- | T#45s | Früheste Zeit, nach deren Ablauf eine neue Verbindung aufgebaut werden kann. |
| tAdsTimeout | TIME | Get, Set | -/- | DEFAULT_ADS_TIMEOUT | Maximale Timeoutzeit für ADS-Dienste, sie darf nicht überschritten werden. |
| tSocketTimeout | TIME | Get, Set | -/- | T#5s | Maximale Timeoutzeit die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| bThrottleMode | BOOL | Get, Set | -/- | TRUE | Aktiviert/deaktiviert den Empfangsdaten-Throttle-Polling-Mode. |
| throttleTimes | T_ThrottleTimes
[► 453] | Get, Set | -/- | Param_Sockets.cDefaultReceiverThrottleTimes | Konfiguration der Zykluszeiten für den Empfangsdaten-Throttle-Polling-Mode. |
| eTraceSocket | E_TraceLevel
[► 449] | Get, Set | -/- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| ipTLS | I_SocketTlsSettingsClass
[► 331] | Get, Set | -/- | 0 | Konfigurationsparameter der TLS-Protokollschicht (Transport Layer Security). Wert Null = nicht konfiguriert. |
| tTpktTimeout | TIME | Get, Set | Geerbt von
FB_TpktClientSettingsClass
[► 205] | T#10s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceTpkt | E_TraceLevel
[► 449] | Get, Set | -/- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| sCalling_T_Selector | STRING | Get
Set | I_Rfc1006ClientSettingsClass
[► 327]
Local | ‘00 00’ | RFC 1006 Parameter: Calling transport address selector (source). |

| Name | Typ | Zugriff | Definitionsart | Initialwert | Beschreibung |
|--------------------|---|---------|--|-----------------------------|--|
| sCalled_T_Selector | STRING | Get | I_Rfc1006ClientSettingsClass [▶ 327] | '00 01' | RFC 1006 Parameter: Called transport address selector (destination). |
| | | Set | Local | | |
| eTpduSize | E_Rfc1006TpduSize [▶ 448] | Get | I_Rfc1006ClientSettingsClass [▶ 327] | E_Rfc1006TpduSize.Size_1024 | Maximale TPDU-Bytelänge. |
| | | Set | Local | | |
| tRfc1006Timeout | TIME | Get | I_Rfc1006ClientSettingsClass [▶ 327] | T#15s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| | | Set | Local | | |
| eTraceRfc1006 | E_TraceLevel [▶ 449] | Get | I_Rfc1006ClientSettingsClass [▶ 327] | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| | | Set | Local | | |

6.3.2.5 FB_Rfc1006ServerSettingsClass

Der Funktionsbaustein FB_Rfc1006ServerSettingsClass dient der Konfiguration der RFC1006-Kommunikationsschicht.

Namensraum: [Tc3_Rfc1006](#) [\[▶ 97\]](#)

Bibliothek: Tc3_Rfc1006 (Tc3_Rfc1006.compiled-library)

Vererbungshierarchie

[FB_SocketServerSettingsClass](#) [\[▶ 201\]](#) -> [FB_TpktServerSettingsClass](#) [\[▶ 207\]](#) -> [FB_Rfc1006ServerSettingsClass](#)

```
FUNCTION_BLOCK FB_Rfc1006ServerSettingsClass EXTENDS FB_TpktServerSettingsClass IMPLEMENTS I_Rfc1006ServerSettingsClass
```

Schnittstellen

| Typ | Beschreibung |
|--|---|
| I_Rfc1006ServerSettingsClass [▶ 328] | Schnittstelle zur Konfiguration der RFC1006-Kommunikationsschicht |

 **Eigenschaften**

| Name | Typ | Zugriff | Definitionsort | Initialwert | Beschreibung |
|--------------------|---|----------|---|---|---|
| bAutoCleanu
p | BOOL | Get, Set | Geerbt von
FB_SocketServerSettings
Class [► 201] | TRUE | Aktiviert/Deaktiviert das automatische Aufräumen/Schließen der unbenutzten Sockets (z.B. nach Programmdownload oder Reset der SPS). |
| bEnable | BOOL | Get, Set | --- | FALSE | Aktiviert/deaktiviert das Herstellen der Serververbindungen. Beim „TRUE“ wird Listen-Befehl aktiviert (horchen und ankommende Verbindungen werden akzeptiert) und beim „FALSE“ abgelehnt/deaktiviert. |
| sSrvNetID | T_AmsNetID
[► 452] | Get, Set | --- | “ | TwinCAT Netzwerkadresse des TwinCAT TCP/IP Servers (TF6310). Beim Leerstring wird die Netzwerkadresse des lokalen TwinCAT Systems verwendet. |
| nLocalPort | UDINT | Get, Set | --- | '127.0.0.1' | TCP/IP Portnummer des Servers. |
| sLocalHost | T_Ipv4Addr
[► 452] | Get, Set | --- | 102 | IPv4-Netzwerkadresse des Servers. |
| tAccept | TIME | Get, Set | --- | T#1s | Wiederholungszykluszeit für die internen TCP/IP-Socket Accept-Befehle. |
| tListen | TIME | Get, Set | --- | T#1s | Wiederholungszykluszeit für den internen TCP/IP-Socket Listen-Befehl. |
| tAdsTimeout | TIME | Get, Set | --- | DEFAULT_A
DS_TIMEOU
T | Maximale Timeoutzeit für ADS-Dienste, sie darf nicht überschritten werden. |
| tSocketTime
out | TIME | Get, Set | --- | T#5s | Maximale Timeoutzeit die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| bThrottleMod
e | BOOL | Get, Set | --- | TRUE | Aktiviert/deaktiviert den Empfangsdaten-Throttle-Polling-Mode. |
| throttleTimes | T_ThrottleTimes
[► 453] | Get, Set | --- | Param_Sock
ets.cDefault
ReceiverThr
ottleTimes | Konfiguration der Zykluszeiten für den Empfangsdaten-Throttle-Polling-Mode. |
| eTraceSocke
t | E_TraceLevel
[► 449] | Get, Set | --- | E_TraceLev
el.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| ipTLS | I_SocketTlsSettin
gsClass [► 331] | Get, Set | --- | 0 | Konfigurationsparameter der TLS-Protokollschicht (Transport Layer Security). Wert Null = nicht konfiguriert. |
| | | | | | |

| Name | Typ | Zugriff | Definitionsart | Initialwert | Beschreibung |
|---------------------|---|------------|---|-----------------------------|--|
| tTpktTimeout | TIME | Get, Set | Geerbt von FB_TpktServerSettingsClass [▶ 207] | T#10s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceTpkt | E_TraceLevel [▶ 449] | Get, Set | --- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| sCalling_T_Selector | STRING | Get
Set | I_Rfc1006ServerSettingsClass [▶ 328]
Local | '00 00' | RFC 1006 Parameter: Calling transport address selector (source). |
| sCalled_T_Selector | STRING | Get
Set | I_Rfc1006ServerSettingsClass [▶ 328]
Local | '00 01' | RFC 1006 Parameter: Called transport address selector (destination). |
| eTpduSize | E_Rfc1006TpduSize [▶ 448] | Get
Set | I_Rfc1006ServerSettingsClass [▶ 328]
Local | E_Rfc1006TpduSize.Size_1024 | Maximale TPDU-Bytelänge. |
| tRfc1006Timeout | TIME | Get
Set | I_Rfc1006ServerSettingsClass [▶ 328]
Local | T#15s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceRfc1006 | E_TraceLevel [▶ 449] | Get
Set | I_Rfc1006ServerSettingsClass [▶ 328]
Local | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |

6.3.2.6 FB_ScsmClientSettingsClass

```

FB_ScsmClientSettingsClass
—services FB\_MmsServiceSupportOptions
—parameters FB\_MmsParameterSupportOptions
—extendedServices FB\_MmsAdditionalSupportOptions
—extendedParameters FB\_MmsAdditionalCBBOptions
    
```

Der Funktionsbaustein `FB_ScsmClientSettingsClass` dient der Konfiguration der SCSM-Kommunikationsschicht.

Namensraum: [Tc3_iec61850_8_1](#) [[▶ 97](#)]

Bibliothek: `Tc3_iec61850_8_1` (`Tc3_iec61850_8_1.compiled-library`)

Vererbungshierarchie

[FB_SocketClientSettingsClass](#) [[▶ 200](#)] -> [FB_TpktClientSettingsClass](#) [[▶ 205](#)] -> [FB_Rfc1006ClientSettingsClass](#) [[▶ 183](#)] -> [FB_UlosiClientSettingsClass](#) [[▶ 210](#)] -> [FB_MmsClientSettingsClass](#) [[▶ 169](#)] -> [FB_ScsmClientSettingsClass](#)

```

FUNCTION_BLOCK FB_ScsmClientSettingsClass EXTENDS FB_MmsClientSettingsClass IMPLEMENTS I_ScsmClientSettingsClass
    
```

➔ Schnittstellen

| Typ | Beschreibung |
|---|--|
| I_ScsmClientSettingsClass [▶ 328] | Schnittstelle zur Konfiguration der SCSM-Kommunikationsschicht |

 **Eingänge**

| Name | Typ | Definitionsort | Beschreibung |
|--------------------|--|--|--|
| services | FB_MmsServiceSupportOptions [▶ 175] | Geerbt von FB_MmsClientSettingsClass [▶ 169] | Konfiguration der unterstützten MMS-Protokolldienste. |
| parameters | FB_MmsParameterSupportOptions [▶ 175] | -//- | Konfiguration der unterstützten MMS-Protokollparameter. |
| extendedServices | FB_MmsAdditionalSupportOptions [▶ 176] | -//- | Konfiguration der zusätzlich unterstützten MMS-Protokolldienste. |
| extendedParameters | FB_MmsAdditionalCBBOptions [▶ 176] | -//- | Konfiguration der unterstützten MMS-CBB-Protokollparameter. |

 **Eigenschaften**

| Name | Typ | Zugriff | Definitionsort | Initialwert | Beschreibung |
|---------------------|--|----------|--|---|--|
| bAutoCleanup | BOOL | Get, Set | Geerbt von FB_SocketClientSettingsClass [▶ 200] | TRUE | Aktiviert/Deaktiviert das automatische Aufräumen/Schließen der unbenutzten Sockets (z.B. nach Programmdownload oder Reset der SPS). |
| sSrvNetID | T_AmsNetID [▶ 452] | Get, Set | --- | '' | TwinCAT Netzwerkadresse des TwinCAT TCP/IP Servers (TF6310). Beim Leerstring wird die Netzwerkadresse des lokalen TwinCAT Systems verwendet. |
| nRemotePort | UDINT | Get, Set | --- | '127.0.0.1' | TCP/IP Portnummer des Servers. |
| sRemoteHost | T_Ipv4Address [▶ 452] | Get, Set | --- | 102 | IPv4-Netzwerkadresse des Servers. |
| tConnect | TIME | Get, Set | --- | T#45s | Früheste Zeit, nach deren Ablauf eine neue Verbindung aufgebaut werden kann. |
| tAdsTimeout | TIME | Get, Set | --- | DEFAULT_ADS_TIMEOUT | Maximale Timeoutzeit für ADS-Dienste, sie darf nicht überschritten werden. |
| tSocketTimeout | TIME | Get, Set | --- | T#5s | Maximale Timeoutzeit die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| bThrottleMode | BOOL | Get, Set | --- | TRUE | Aktiviert/deaktiviert den Empfangsdaten-Throttle-Polling-Mode. |
| throttleTimes | T_ThrottleTimes [▶ 453] | Get, Set | --- | Param_Sockets.cDefaultReceiverThrottleTimes | Konfiguration der Zykluszeiten für den Empfangsdaten-Throttle-Polling-Mode. |
| eTraceSocket | E_TraceLevel [▶ 449] | Get, Set | --- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| ipTLS | I_SocketTlsSettingsClass [▶ 331] | Get, Set | --- | 0 | Konfigurationsparameter der TLS-Protokollschicht (Transport Layer Security). Wert Null = nicht konfiguriert. |
| tTpktTimeout | TIME | Get, Set | Geerbt von FB_TpktClientSettingsClass [▶ 205] | T#10s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceTpkt | E_TraceLevel [▶ 449] | Get, Set | --- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| sCalling_T_Selector | STRING | Get, Set | Geerbt von FB_Rfc1006ClientSettingsClass [▶ 183] | '00 00' | RFC 1006 Parameter: Calling transport address selector (source). |
| sCalled_T_Selector | STRING | Get, Set | --- | '00 01' | RFC 1006 Parameter: Called transport address selector (destination). |

| Name | Typ | Zugriff | Definitionsort | Initialwert | Beschreibung |
|-----------------------|------------------------------|---------|--|-----------------------------|--|
| eTpduSize | E_Rfc1006TpduSize
[► 448] | Get,Set | --- | E_Rfc1006TpduSize.Size_1024 | Maximale TPDU-Bytelänge. |
| tRfc1006Timeout | TIME | Get,Set | --- | T#15s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceRfc1006 | E_TraceLevel [► 449] | Get,Set | --- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| sCalling_S_Selector | STRING | Get,Set | Geerbt von <u>FB_UlosiClientSettingsClass</u>
[► 210] | '00 01' | Calling-S-selector. |
| sCalled_S_Selector | STRING | Get,Set | --- | '00 01' | Called-S-selector. |
| nSessionRequirement | WORD | Get,Set | --- | 2 | Session requirements. |
| sCalling_P_Selector | STRING | Get,Set | --- | '00 00 00 01' | Calling-P-selector. |
| sCalled_P_Selector | STRING | Get,Set | --- | '00 00 00 01' | Called-P-selector. |
| nProtocolVersion | BYTE | Get,Set | --- | 1 | Protocol version. |
| sContext_Name | STRING | Get,Set | --- | '1.0.9506.2.3' | Application context name (object identifier, dotted presentation), MMS. |
| sCalled_AP_Title | STRING | Get,Set | --- | '1.1.1.999.1' | Called-AP-title. |
| nCalled_AE_Qualifier | UDINT | Get,Set | --- | 12 | Called-AE-qualifier. |
| nCalled_AP_InvID | UDINT | Get,Set | --- | 0 | Called-AP-invocation-identifier. |
| nCalled_AE_InvID | UDINT | Get,Set | --- | 0 | Called-AE-invocation-identifier. |
| sCalling_AP_Title | STRING | Get,Set | --- | '1.1.1.999' | Calling-AP-title. |
| nCalling_AE_Qualifier | UDINT | Get,Set | --- | 12 | Calling-AE-qualifier. |
| nCalling_AP_InvID | UDINT | Get,Set | --- | 0 | Calling-AP-invocation-identifier. |
| nCalling_AE_InvID | UDINT | Get,Set | --- | 0 | Calling-AE-invocation-identifier. |
| nRequirements | BYTE | Get,Set | --- | 0 | Requirements (1 == authentication). |
| sMechanism_Name | STRING | Get,Set | --- | '2.2.3.1' | Authentication mechanism name. |
| sAuthent_Value | STRING | Get,Set | --- | '' | Authentication value (password). |
| tUlosiTimeout | TIME | Get,Set | --- | T#20s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |

| Name | Typ | Zu-griff | Definitionsor-t | Initialwert | Beschreibung |
|-------------------------|--|-------------|--|-------------------|--|
| eTraceUlosi | E_TraceLev
el [► 449] | Get,S
et | --- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| nLocalDetail | T_MmsInteger32
[► 478] | Get,S
et | Geerbt von
FB_MmsClientSettingsClass
[► 169] | 65000 | MMS Initiate-RequestPDU-Parameter: localDetailCalling. |
| nMaxServOutCalling | T_MmsInteger16
[► 477] | Get,S
et | --- | 10 | MMS Initiate-RequestPDU-Parameter:
proposedMaxServOutstandingCalling |
| nMaxServOutCalled | T_MmsInteger16
[► 477] | Get,S
et | --- | 10 | MMS Initiate-RequestPDU-Parameter:
proposedMaxServOutstandingCalled. |
| nNestingLevel | T_MmsInteger8
[► 477] | Get,S
et | --- | 5 | MMS Initiate-RequestPDU-Parameter:
proposedDataStructureNestingLevel. |
| nMmsVersionNumber | T_MmsInteger16
[► 477] | Get,S
et | --- | 16#0001 | MMS initRequestDetail-Parameter:
proposedVersionNumber. |
| ipParameters | I_MmsParameterSupportOptions
[► 325] | Get,S
et | --- | - | Schnittstellenzeiger vom Objekt zur Konfiguration der MMS initRequestDetail-Parameter:
proposedParameterCBB. |
| ipServices | I_MmsServiceSupportOptions
[► 324] | Get,S
et | --- | - | Schnittstellenzeiger vom Objekt zur Konfiguration der MMS initRequestDetail-Parameter:
servicesSupportedCalling. |
| ipExtendedServices | I_MmsAdditionalSupportOptions
[► 325] | Get,S
et | --- | - | Schnittstellenzeiger vom Objekt zur Konfiguration der MMS initRequestDetail-Parameter:
additionalSupportedCalling.. |
| ipExtendedParameters | I_MmsAdditionalCBBOptions
[► 326] | Get,S
et | --- | - | Schnittstellenzeiger vom Objekt zur Konfiguration der MMS initRequestDetail-Parameter:
additionalCbbSupportedCalling. |
| sPrivilegeClassIdentity | T_MmsVisibleString
[► 478] | Get,S
et | --- | “ | MMS-initRequestDetail-Parameter:
privilegeClassIdentityCalled. |
| tMmsTimeout | TIME | Get,S
et | --- | T#25s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceMms | E_TraceLevel [► 449] | Get,S
et | --- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |

| Name | Typ | Zu-griff | Definitionsor | Initialwert | Beschreibung |
|-------------------|---|----------|---|-------------------|--|
| bAutoInitDataSets | BOOL | Get | I_ScsmClientSettingsClass [▶ 328] | TRUE | Aktiviert/deaktiviert die automatische Initialisierung der Datasets/Dataset-Member. |
| | | Set | Local | | |
| eScsmEdition | E_ScsmEdition [▶ 448] | Get | I_ScsmClientSettingsClass [▶ 328] | E_ScsmEdition.Ed2 | IEC 61850 Datenmodell Edition (1 oder 2) |
| | | Set | Local | | |
| tScsmTimeout | TIME | Get | I_ScsmClientSettingsClass [▶ 328] | T#30s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| | | Set | Local | | |
| eTraceScsm | T_TraceLevel [▶ 449] | Get | I_ScsmClientSettingsClass [▶ 328] | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| | | Set | Local | | |

6.3.2.7 FB_ScsmServerSettingsClass

Der Funktionsbaustein FB_ScsmServerSettingsClass dient der Konfiguration der SCSM-Kommunikationsschicht.

Namensraum: [Tc3_iec61850_8_1](#) [[▶ 97](#)]

Bibliothek: [Tc3_iec61850_8_1](#) ([Tc3_iec61850_8_1.compiled-library](#))

Vererbungshierarchie

[FB_SocketServerSettingsClass](#) [[▶ 201](#)] -> [FB_TpktServerSettingsClass](#) [[▶ 207](#)] -> [FB_Rfc1006ServerSettingsClass](#) [[▶ 186](#)] -> [FB_UlosiServerSettingsClass](#) [[▶ 214](#)] -> [FB_MmsServerSettingsClass](#) [[▶ 177](#)] -> [FB_ScsmServerSettingsClass](#)

```
FUNCTION_BLOCK FB_ScsmServerSettingsClass EXTENDS FB_MmsServerSettingsClass IMPLEMENTS I_ScsmServerSettingsClass
```

Schnittstellen

| Typ | Beschreibung |
|---|--|
| I_ScsmServerSettingsClass [▶ 329] | Schnittstelle zur Konfiguration der SCSM-Kommunikationsschicht |

Eingänge

| Name | Typ | Definitionsor | Beschreibung |
|--------------------|--|--|--|
| services | FB_MmsServiceSupportOptions [▶ 175] | Geerbt von FB_MmsServerSettingsClass [▶ 177] | Konfiguration der unterstützten MMS-Protokolldienste. |
| parameters | FB_MmsParameterSupportOptions [▶ 175] | -/- | Konfiguration der unterstützten MMS-Protokollparameter. |
| extendedServices | FB_MmsAdditionalSupportOptions [▶ 176] | -/- | Konfiguration der zusätzlich unterstützten MMS-Protokolldienste. |
| extendedParameters | FB_MmsAdditionalCBBOptions [▶ 176] | -/- | Konfiguration der unterstützten MMS-CBB-Protokollparameter. |

 **Eigenschaften**

| Name | Typ | Zugriff | Definitionsort | Initialwert | Beschreibung |
|----------------|--|----------|---|---|---|
| bAutoCleanup | BOOL | Get, Set | Geerbt von FB_SocketServerSettingsClass [▶ 201] | TRUE | Aktiviert/Deaktiviert das automatische Aufräumen/Schließen der unbenutzten Sockets (z.B. nach Programmdownload oder Reset der SPS). |
| bEnable | BOOL | Get, Set | --- | FALSE | Aktiviert/deaktiviert das Herstellen der Serververbindungen. Beim „TRUE“ wird Listen-Befehl aktiviert (horchen und ankommende Verbindungen werden akzeptiert) und beim „FALSE“ abgelehnt/deaktiviert. |
| sSrvNetID | T_AmsNetID [▶ 452] | Get, Set | --- | “ | TwinCAT Netzwerkadresse des TwinCAT TCP/IP Servers (TF6310). Beim Leerstring wird die Netzwerkadresse des lokalen TwinCAT Systems verwendet. |
| nLocalPort | UDINT | Get, Set | --- | ‘127.0.0.1’ | TCP/IP Portnummer des Servers. |
| sLocalHost | T_Ipv4Addr [▶ 452] | Get, Set | --- | 102 | IPv4-Netzwerkadresse des Servers. |
| tAccept | TIME | Get, Set | --- | T#1s | Wiederholungszykluszeit für die internen TCP/IP-Socket Accept-Befehle. |
| tListen | TIME | Get, Set | --- | T#1s | Wiederholungszykluszeit für den internen TCP/IP-Socket Listen-Befehl. |
| tAdsTimeout | TIME | Get, Set | --- | DEFAULT_ADS_TIMEOUT | Maximale Timeoutzeit für ADS-Dienste, sie darf nicht überschritten werden. |
| tSocketTimeout | TIME | Get, Set | --- | T#5s | Maximale Timeoutzeit die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| bThrottleMode | BOOL | Get, Set | --- | TRUE | Aktiviert/deaktiviert den Empfangsdaten-Throttle-Polling-Mode. |
| throttleTimes | T_ThrottleTimes [▶ 453] | Get, Set | --- | Param_Sockets.cDefaultReceiverThrottleTimes | Konfiguration der Zykluszeiten für den Empfangsdaten-Throttle-Polling-Mode. |
| eTraceSocket | E_TraceLevel [▶ 449] | Get, Set | --- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| ipTLS | I_SocketTlsSettingsClass [▶ 331] | Get, Set | --- | 0 | Konfigurationsparameter der TLS-Protokollschicht (Transport Layer Security). Wert Null = nicht konfiguriert. |
| tTpktTimeout | TIME | Get, Set | Geerbt von FB_TpktServerSettingsClass [▶ 207] | T#10s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceTpkt | E_TraceLevel [▶ 449] | Get, Set | --- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |

| Name | Typ | Zu-griff | Definitionsort | Initialwert | Beschreibung |
|-----------------------|---|----------|--|-----------------------------|--|
| sCalling_T_Selector | STRING | Get,Set | Geerbt von FB_Rfc1006ServerSettingsClass [▶ 186] | '00 00' | RFC 1006 Parameter: Calling transport address selector (source). |
| sCalled_T_Selector | STRING | Get,Set | --- | '00 01' | RFC 1006 Parameter: Called transport address selector (destination). |
| eTpduSize | E_Rfc1006TpduSize [▶ 448] | Get,Set | --- | E_Rfc1006TpduSize.Size_1024 | Maximale TPDU-Bytelänge. |
| tRfc1006Timeout | TIME | Get,Set | --- | T#15s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceRfc1006 | E_TraceLevel [▶ 449] | Get,Set | --- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| sCalling_S_Selector | STRING | Get,Set | Geerbt von FB_UlosiServerSettingsClass [▶ 214] | '00 01' | Calling-S-selector. |
| sCalled_S_Selector | STRING | Get,Set | --- | '00 01' | Called-S-selector. |
| nSessionRequirement | WORD | Get,Set | --- | 2 | Session requirements. |
| sCalling_P_Selector | STRING | Get,Set | --- | '00 00 00 01' | Calling-P-selector. |
| sCalled_P_Selector | STRING | Get,Set | --- | '00 00 00 01' | Called-P-selector. |
| nProtocolVersion | BYTE | Get,Set | --- | 1 | Protocol version. |
| sContext_Name | STRING | Get,Set | --- | '1.0.9506.2.3' | Applicaton context name (object identifier, dotted presentation), MMS. |
| sCalled_AP_Title | STRING | Get,Set | --- | '1.1.1.999.1' | Called-AP-title. |
| nCalled_AE_Qualifier | UDINT | Get,Set | --- | 12 | Called-AE-qualifier. |
| nCalled_AP_InvID | UDINT | Get,Set | --- | 0 | Called-AP-invocation-identifier. |
| nCalled_AE_InvID | UDINT | Get,Set | --- | 0 | Called-AE-invocation-identifier. |
| sCalling_AP_Title | STRING | Get,Set | --- | '1.1.1.999' | Calling-AP-title. |
| nCalling_AE_Qualifier | UDINT | Get,Set | --- | 12 | Calling-AE-qualifier. |
| nCalling_AP_InvID | UDINT | Get,Set | --- | 0 | Calling-AP-invocation-identifier. |
| nCalling_AE_InvID | UDINT | Get,Set | --- | 0 | Calling-AE-invocation-identifier. |

| Name | Typ | Zu-griff | Definitionsor | Initialwert | Beschreibung |
|-------------------------|---------------------------------------|----------|--|-------------------|---|
| nRequirements | BYTE | Get,Set | -/- | 0 | Requirements (1 == authentication). |
| sMechanism_Name | STRING | Get,Set | -/- | '2.2.3.1' | Authentication mechanism name. |
| sAuthent_Value | STRING | Get,Set | -/- | '' | Authentication value (password). |
| tUlosiTimeout | TIME | Get,Set | -/- | T#20s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceUlosi | E_TraceLevel [▶ 449] | Get,Set | -/- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| nLocalDetail | T_MmsInteger32 [▶ 478] | Get,Set | Geerbt von FB_MmsServerSettingsClass [▶ 177] | 65000 | MMS Initiate-RequestPDU-Parameter: localDetailCalling. |
| nMaxServOutCalling | T_MmsInteger16 [▶ 477] | Get,Set | -/- | 10 | MMS Initiate-RequestPDU-Parameter: proposedMaxServOutstandingCalling |
| nMaxServOutCalled | T_MmsInteger16 [▶ 477] | Get,Set | -/- | 10 | MMS Initiate-RequestPDU-Parameter: proposedMaxServOutstandingCalled. |
| nNestingLevel | T_MmsInteger8 [▶ 477] | Get,Set | -/- | 5 | MMS Initiate-RequestPDU-Parameter: proposedDataStructureNestingLevel. |
| nMmsVersionNumber | T_MmsInteger16 [▶ 477] | Get,Set | -/- | 16#0001 | MMS initRequestDetail-Parameter: proposedVersionNumber. |
| ipParameters | I_MmsParameterSupportOptions [▶ 325] | Get,Set | -/- | - | Schnittstellenzeiger vom Objekt zur Konfiguration der MMS initRequestDetail-Parameter: proposedParameterCBB. |
| ipServices | I_MmsServiceSupportOptions [▶ 324] | Get,Set | -/- | - | Schnittstellenzeiger vom Objekt zur Konfiguration der MMS initRequestDetail-Parameter: servicesSupportedCalling. |
| ipExtendedServices | I_MmsAdditionalSupportOptions [▶ 325] | Get,Set | -/- | - | Schnittstellenzeiger vom Objekt zur Konfiguration der MMS initRequestDetail-Parameter: additionalSupportedCalling.. |
| ipExtendedParameters | I_MmsAdditionalCBBOptions [▶ 326] | Get,Set | -/- | - | Schnittstellenzeiger vom Objekt zur Konfiguration der MMS initRequestDetail-Parameter: additionalCbbSupportedCalling. |
| sPrivilegeClassIdentity | T_MmsVisibleString [▶ 478] | Get,Set | -/- | '' | MMS-initRequestDetail-Parameter: privilegeClassIdentityCalled. |

| Name | Typ | Zugriff | Definitionsort | Initialwert | Beschreibung |
|-------------------|---------------------------------------|------------|--|-------------------|--|
| tMmsTimeout | TIME | Get, Set | -/- | T#25s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceMms | E_TraceLevel [▶ 449] | Get, Set | -/- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| bAutoInitDataSets | BOOL | Get
Set | I_ScsmServerSettingsClass [▶ 329]
Local | TRUE | Aktiviert/deaktiviert die automatische Initialisierung der Datasets/Dataset-Member. |
| eScsmEdition | E_ScsmEdition [▶ 448] | Get
Set | I_ScsmServerSettingsClass [▶ 329]
Local | E_ScsmEdition.Ed2 | IEC 61850 Datenmodell Edition (1 oder 2) |
| tScsmTimeout | TIME | Get
Set | I_ScsmServerSettingsClass [▶ 329]
Local | T#30s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceScsm | T_TraceLevel [▶ 449] | Get
Set | I_ScsmServerSettingsClass [▶ 329]
Local | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |

6.3.2.8 FB_SocketClientSettingsClass

Der Funktionsbaustein FB_SocketClientSettingsClass dient der Konfiguration der TCP/IP-Kommunikationsschicht.

Namensraum: Tc3_Sockets [\[▶ 97\]](#)

Bibliothek: Tc3_Sockets (Tc3_Sockets.compiled-library)

Vererbungshierarchie

FB_SocketClientSettingsClass

FUNCTION_BLOCK FB_SocketClientSettingsClass IMPLEMENTS I_SocketClientSettingsClass

Schnittstellen

| Typ | Beschreibung |
|---|---|
| I_SocketClientSettingsClass [▶ 329] | Schnittstelle zur Konfiguration der TCP/IP-Kommunikationsschicht. |

 **Eigenschaften**

| Name | Typ | Zugriff | Definitionsart | Initialwert | Beschreibung |
|----------------|--|---------|---|---|--|
| bAutoCleanUp | BOOL | Get | I_SocketClientSettingsClass [▶ 329] | TRUE | Aktiviert/Deaktiviert das automatische Aufräumen/Schließen der unbenutzten Sockets (z.B. nach Programmdownload oder Reset der SPS). |
| | | Set | Local | | |
| sSrvNetID | T_AmsNetID [▶ 452] | Get | I_SocketClientSettingsClass [▶ 329] | “ | TwinCAT Netzwerkadresse des TwinCAT TCP/IP Servers (TF6310). Beim Leerstring wird die Netzwerkadresse des lokalen TwinCAT Systems verwendet. |
| | | Set | Local | | |
| nRemotePort | UDINT | Get | I_SocketClientSettingsClass [▶ 329] | ‘127.0.0.1’ | TCP/IP Portnummer des Servers. |
| | | Set | Local | | |
| sRemoteHost | T_Ipv4Address [▶ 452] | Get | I_SocketClientSettingsClass [▶ 329] | 102 | IPv4-Netzwerkadresse des Servers. |
| | | Set | Local | | |
| tConnect | TIME | Get | I_SocketClientSettingsClass [▶ 329] | T#45s | Früheste Zeit, nach deren Ablauf eine neue Verbindung aufgebaut werden kann. |
| | | Set | Local | | |
| tAdsTimeout | TIME | Get | I_SocketClientSettingsClass [▶ 329] | DEFAULT_ADS_TIMEOUT | Maximale Timeoutzeit für ADS-Dienste, sie dürfen nicht überschritten werden. |
| | | Set | Local | | |
| tSocketTimeout | TIME | Get | I_SocketClientSettingsClass [▶ 329] | T#5s | Maximale Timeoutzeit die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| | | Set | Local | | |
| bThrottleMode | BOOL | Get | I_SocketClientSettingsClass [▶ 329] | TRUE | Aktiviert/deaktiviert den Empfangsdaten-Throttle-Polling-Mode. |
| | | Set | Local | | |
| throttleTimes | T_ThrottleTimes [▶ 453] | Get | I_SocketClientSettingsClass [▶ 329] | Param_Sockets.cDefaultReceiverThrottleTimes | Konfiguration der Zykluszeiten für den Empfangsdaten-Throttle-Polling-Mode. |
| | | Set | Local | | |
| eTraceSocket | E_TraceLevel [▶ 449] | Get | I_SocketClientSettingsClass [▶ 329] | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| | | Set | Local | | |
| ipTLS | I_SocketTlsSettingsClass [▶ 331] | Get | I_SocketTlsSettingsClass [▶ 331] | 0 | Konfigurationsparameter der TLS-Protokollschicht (Transport Layer Security). Wert Null = nicht konfiguriert. |
| | | Set | Local | | |

6.3.2.9 FB_SocketServerSettingsClass

Der Funktionsbaustein FB_SocketServerSettingsClass dient der Konfiguration der TCP/IP-Kommunikationsschicht.

Namensraum: [Tc3_Sockets](#) [[▶ 97](#)]


Bibliothek: Tc3_Sockets (Tc3_Sockets.compiled-library)

Vererbungshierarchie

FB_SocketServerSettingsClass

```
FUNCTION_BLOCK FB_SocketServerSettingsClass IMPLEMENTS I_SocketServerSettingsClass
```

Schnittstellen

| Typ | Beschreibung |
|---|---|
| I_SocketServerSettingsClass  330 | Schnittstelle zur Konfiguration der TCP/IP-Kommunikationsschicht. |

 **Eigenschaften**

| Name | Typ | Zugriff | Definitionsart | Initialwert | Beschreibung |
|----------------|--|---------|---|---|---|
| bAutoCleanup | BOOL | Get | I_SocketServerSettingsClass [▶ 330] | TRUE | Aktiviert/Deaktiviert das automatische Aufräumen/Schließen der unbenutzten Sockets (z. B. nach Programmdownload oder Reset der SPS). |
| | | Set | Local | | |
| bEnable | BOOL | Get | I_SocketServerSettingsClass [▶ 330] | FALSE | Aktiviert/deaktiviert das Herstellen der Serververbindungen. Beim „TRUE“ wird Listen-Befehl aktiviert (horchen und ankommende Verbindungen werden akzeptiert) und beim „FALSE“ abgelehnt/deaktiviert. |
| | | Set | Local | | |
| sSrvNetID | T_AmsNetID [▶ 452] | Get | I_SocketServerSettingsClass [▶ 330] | “ | TwinCAT Netzwerkadresse des TwinCAT TCP/IP Servers (TF6310). Beim Leerstring wird die Netzwerkadresse des lokalen TwinCAT Systems verwendet. |
| | | Set | Local | | |
| nLocalPort | UDINT | Get | I_SocketServerSettingsClass [▶ 330] | ‘127.0.0.1’ | TCP/IP Portnummer des Servers. |
| | | Set | Local | | |
| sLocalHost | T_Ipv4Addr [▶ 452] | Get | I_SocketServerSettingsClass [▶ 330] | 102 | IPv4-Netzwerkadresse des Servers. |
| | | Set | Local | | |
| tAccept | TIME | Get | I_SocketServerSettingsClass [▶ 330] | T#1s | Wiederholungszykluszeit für die internen TCP/IP-Socket Accept-Befehle. |
| | | Set | Local | | |
| tListen | TIME | Get | I_SocketServerSettingsClass [▶ 330] | T#1s | Wiederholungszykluszeit für den internen TCP/IP-Socket Listen-Befehl. |
| | | Set | Local | | |
| tAdsTimeout | TIME | Get | I_SocketServerSettingsClass [▶ 330] | DEFAULT_ADS_TIMEOUT | Maximale Timeoutzeit für ADS-Dienste, sie darf nicht überschritten werden. |
| | | Set | Local | | |
| tSocketTimeout | TIME | Get | I_SocketServerSettingsClass [▶ 330] | T#5s | Maximale Timeoutzeit die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| | | Set | Local | | |
| bThrottleMode | BOOL | Get | I_SocketServerSettingsClass [▶ 330] | TRUE | Aktiviert/deaktiviert den Empfangsdaten-Throttle-Polling-Mode. |
| | | Set | Local | | |
| throttleTimes | T_ThrottleTimes [▶ 453] | Get | I_SocketServerSettingsClass [▶ 330] | Param_Sockets.cDefaultReceiverThrottleTimes | Konfiguration der Zykluszeiten für den Empfangsdaten-Throttle-Polling-Mode. |
| | | Set | Local | | |
| eTraceSocket | E_TraceLevel [▶ 449] | Get | I_SocketServerSettingsClass [▶ 330] | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| | | Set | Local | | |
| ipTLS | I_SocketTlsSettingsClass [▶ 331] | Get | I_SocketTlsSettingsClass [▶ 331] | 0 | Konfigurationsparameter der TLS-Protokollschicht (Transport Layer Security). Wert Null = nicht konfiguriert. |
| | | Set | Local | | |

6.3.2.10 FB_SocketTlsSettingsClass

Der Funktionsbaustein FB_SocketTlsSettingsClass dient der Konfiguration der TLS (Transport Layer Security) Kommunikationsschicht. TLS ermöglicht eine sichere IEC 61850 Client-Server-Kommunikation.

Namensraum: Tc3_Sockets [[▶ 97](#)]

Bibliothek: Tc3_Sockets (Tc3_Sockets.compiled-library)

Vererbungshierarchie

FB_SocketTlsSettingsClass

FUNCTION_BLOCK FB_SocketTlsSettingsClass IMPLEMENTS I_SocketTlsSettingsClass

Schnittstellen

| Typ | Beschreibung |
|--|-------------------------------------|
| I_SocketTlsSettingsClass [▶ 331] | Schnittstelle zur TLS-Konfiguration |

Methoden

| Name | Beschreibung |
|---|--|
| AddCa [▶ 332] | Speichert den Dateipfad des CA-Zertifikats in den TLS-Konfigurationseinstellungen |
| AddCert [▶ 333] | Speichert den Dateipfad des Client- oder Server-Zertifikats in den TLS-Konfigurationseinstellungen |
| AddCrl [▶ 333] | Speichert den Dateipfad der Zertifikatsperrliste (Certificate Revocation List, CRL) in den TLS-Konfigurationseinstellungen |
| AddPsk [▶ 333] | Speichert den PSK (Pre-Shared Key) und PSK-Identity-String in den TLS-Konfigurationseinstellungen |
| Reset [▶ 334] | Setzt alle TLS-Konfigurationseinstellungen zurück |
| SetConnectFlags [▶ 334] | Speichert zusätzliche, optionale TLS-Konfigurationseinstellungen des Clients |
| SetListenFlags [▶ 334] | Speichert zusätzliche, optionale TLS-Konfigurationseinstellungen des Servers |

 **Eigenschaften**

| Name | Typ | Zugriff | Definitionsart | Initialwert | Beschreibung |
|--------------|---|---------|--|-------------------------|---|
| nSecurePort | UDINT | Get | I_SocketTlsSettingsClass [▶ 331] | 3782 | Sichere ISO TP0 (IANA) Portnummer für die TLS-Kommunikation |
| | | Set | Local | | |
| connectFlags | ST_TlsConnectFlags | Get | I_SocketTlsSettingsClass [▶ 331] | DEFAULT_TLSCONNECTFLAGS | Zusätzliche TLS-Konfigurationseinstellungen des Clients |
| listenFlags | ST_TlsListenFlags | Get | I_SocketTlsSettingsClass [▶ 331] | DEFAULT_TLSLISTENFLAGS | Zusätzliche TLS-Konfigurationseinstellungen des Servers |
| pskKey | PVOID | Get | I_SocketTlsSettingsClass [▶ 331] | 0 | Zeigt auf das erste Byte des PSK-Schlüssels |
| pskKeyLen | UDINT(0..TCPADSS_TLS_MAX_PSK_KEY_SIZE) | Get | I_SocketTlsSettingsClass [▶ 331] | 0 | Bytelänge des PSK-Schlüssels |
| sCaPath | STRING(TCPADSS_TLS_CERTIFICATE_PATH_SIZE) | Get | I_SocketTlsSettingsClass [▶ 331] | " | Dateipfad des CA-Zertifikats |
| sCertPath | STRING(TCPADSS_TLS_CERTIFICATE_PATH_SIZE) | Get | I_SocketTlsSettingsClass [▶ 331] | " | Dateipfad des Client- oder Server-Zertifikats |
| sCrlPath | STRING(TCPADSS_TLS_CERTIFICATE_PATH_SIZE) | Get | I_SocketTlsSettingsClass [▶ 331] | " | Dateipfad der Zertifikatsperrliste (Certificate Revocation List, CRL) |
| sIdentity | STRING(TCPADSS_TLS_PSK_IDENTITY_SIZE) | Get | I_SocketTlsSettingsClass [▶ 331] | " | Identity-String des PSK-Schlüssels |
| sKeyPath | STRING(TCPADSS_TLS_CERTIFICATE_PATH_SIZE) | Get | I_SocketTlsSettingsClass [▶ 331] | " | Dateipfad des privaten Schlüssels |
| sKeyPwd | STRING(TCPADSS_TLS_KEY_PASSWORD_SIZE) | Get | I_SocketTlsSettingsClass [▶ 331] | " | Passwort-String des Zertifikats |

6.3.2.11 FB_TpktClientSettingsClass

Der Funktionsbaustein FB_TpktClientSettingsClass dient der Konfiguration der TPKT-Kommunikationsschicht.

Namensraum: [Tc3_Tpkt](#) [[▶ 97](#)]

Bibliothek: Tc3_Tpkt (Tc3_Tpkt.compiled-library)

Vererbungshierarchie

[FB_SocketClientSettingsClass](#) [[▶ 200](#)] -> [FB_TpktClientSettingsClass](#)

```
FUNCTION_BLOCK FB_TpktClientSettingsClass EXTENDS FB_SocketClientSettingsClass IMPLEMENTS I_TpktClientSettingsClass
```

 **Schnittstellen**

| Typ | Beschreibung |
|---|---|
| I_TpktClientSettingsClass [▶ 334] | Schnittstelle zur Konfiguration der TPKT-Kommunikationsschicht. |

 **Eigenschaften**

| Name | Typ | Zu-griff | Definitions-ort | Initialwert | Beschreibung |
|----------------|--|----------|---|---|--|
| bAutoClean-up | BOOL | Get, Set | Geerbt von FB_SocketClientSettingsClass [▶ 200] | TRUE | Aktiviert/Deaktiviert das automatische Aufräumen/Schließen der unbenutzten Sockets (z. B. nach Programmdownload oder Reset der SPS). |
| sSrvNetID | T_AmsNetID [▶ 452] | Get, Set | -/- | “ | TwinCAT Netzwerkadresse des TwinCAT TCP/IP Servers (TF6310). Beim Leerstring wird die Netzwerkadresse des lokalen TwinCAT Systems verwendet. |
| nRemotePort | UDINT | Get, Set | -/- | ‘127.0.0.1’ | TCP/IP Portnummer des Servers. |
| sRemoteHost | T_Ipv4Addr [▶ 452] | Get, Set | -/- | 102 | IPv4-Netzwerkadresse des Servers. |
| tConnect | TIME | Get, Set | -/- | T#45s | Früheste Zeit, nach deren Ablauf eine neue Verbindung aufgebaut werden kann. |
| tAdsTimeout | TIME | Get, Set | -/- | DEFAULT_ADS_TIMEOUT | Maximale Timeoutzeit für ADS-Dienste, sie darf nicht überschritten werden. |
| tSocketTimeout | TIME | Get, Set | -/- | T#5s | Maximale Timeoutzeit die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| bThrottleMode | BOOL | Get, Set | -/- | TRUE | Aktiviert/deaktiviert den Empfangsdaten-Throttle-Polling-Mode. |
| throttleTimes | T_ThrottleTimes [▶ 453] | Get, Set | -/- | Param_Sockets.cDefaultReceiverThrottleTimes | Konfiguration der Zykluszeiten für den Empfangsdaten-Throttle-Polling-Mode. |
| eTraceSocket | E_TraceLevel [▶ 449] | Get, Set | -/- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| ipTLS | I_SocketTlsSettingsClass [▶ 331] | Get, Set | -/- | 0 | Konfigurationsparameter der TLS-Protokollschicht (Transport Layer Security). Wert Null = nicht konfiguriert. |
| tTpktTimeout | TIME | Get | I_TpktClientSettingsClass [▶ 334] | T#10s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| | | Set | Local | | |
| eTraceTpkt | E_TraceLevel [▶ 449] | Get | I_TpktClientSettingsClass [▶ 334] | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| | | Set | Local | | |

6.3.2.12 FB_TpktServerSettingsClass

Der Funktionsbaustein FB_TpktServerSettingsClass dient der Konfiguration der TPKT-Kommunikationsschicht.

Namensraum: Tc3_Tpkt [[▶ 97](#)]

Bibliothek: Tc3_Tpkt (Tc3_Tpkt.compiled-library)

Vererbungshierarchie

[FB_SocketServerSettingsClass \[\[▶ 201\]\(#\)\]](#) -> FB_TpktServerSettingsClass

```
FUNCTION_BLOCK FB_TpktServerSettingsClass EXTENDS FB_SocketServerSettingsClass IMPLEMENTS I_TpktServerSettingsClass
```

Schnittstellen

| Typ | Beschreibung |
|---|---|
| I_TpktServerSettingsClass [▶ 335] | Schnittstelle zur Konfiguration der TPKT-Kommunikationsschicht. |

 **Eigenschaften**

| Name | Typ | Zu-griff | Definitionsort | Initialwert | Beschreibung |
|--------------------|--|----------------|---|---|--|
| bAutoCleanu
p | BOOL | Get,
Set | Geerbt von
FB_SocketServ
erSettingsClas
s_ [▶ 201] | TRUE | Aktiviert/Deaktiviert das
automatische Aufräumen/
Schließen der unbenutzten
Sockets (z. B. nach
Programmdownload oder Reset
der SPS). |
| bEnable | BOOL | Get,
Set | -//- | FALSE | Aktiviert/deaktiviert das
Herstellen der
Serververbindungen. Beim
„TRUE“ wird Listen-Befehl
aktiviert (horchen und
ankommende Verbindungen
werden akzeptiert) und beim
„FALSE“ abgelehnt/deaktiviert. |
| sSrvNetID | T_AmsNetID
 [▶ 452] | Get,
Set | -//- | “ | TwinCAT Netzwerkadresse des
TwinCAT TCP/IP Servers
(TF6310). Beim Leerstring wird
die Netzwerkadresse des
lokalen TwinCAT Systems
verwendet. |
| nLocalPort | UDINT | Get,
Set | -//- | '127.0.0.1' | TCP/IP Portnummer des
Servers. |
| sLocalHost | T_Ipv4Addr
 [▶ 452] | Get,
Set | -//- | 102 | IPv4-Netzwerkadresse des
Servers. |
| tAccept | TIME | Get,
Set | -//- | T#1s | Wiederholungszykluszeit für die
internen TCP/IP-Socket Accept-
Befehle. |
| tListen | TIME | Get,
Set | -//- | T#1s | Wiederholungszykluszeit für den
internen TCP/IP-Socket Listen-
Befehl. |
| tAdsTimeout | TIME | Get,
Set | -//- | DEFAULT_ADS_TIMEO
UT | Maximale Timeoutzeit für ADS-
Dienste, sie darf nicht
überschritten werden. |
| tSocketTime
out | TIME | Get,
Set | -//- | T#5s | Maximale Timeoutzeit die bei
der Ausführung der bestätigten
Dienste nicht überschritten
werden darf. |
| bThrottleMod
e | BOOL | Get,
Set | -//- | TRUE | Aktiviert/deaktiviert den
Empfangsdaten-Throttle-Polling-
Mode. |
| throttleTimes | T_ThrottleTi
mes [▶ 453] | Get,
Set | -//- | Param_Sockets.cDefaul
tReceiverThrottleTimes | Konfiguration der Zykluszeiten
für den Empfangsdaten-Throttle-
Polling-Mode. |
| eTraceSocke
t | E_TraceLevel
 [▶ 449] | Get,
Set | -//- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe
der Protokollierung der
Fehlermeldungen und
Logmeldungen. |
| ipTLS | I_SocketTlsS
ettingsClass
 [▶ 331] | Get,
Set | -//- | 0 | Konfigurationsparameter der
TLS-Protokollschicht (Transport
Layer Security). Wert Null = nicht
konfiguriert. |
| tTpktTimeout | TIME | Get

Set | I_TpktServerSe
ttingsClass
 [▶ 335]

Local | T#10s | Maximale Zeit, die bei der
Ausführung der bestätigten
Dienste nicht überschritten
werden darf. |

| Name | Typ | Zu-griff | Definitionsort | Initialwert | Beschreibung |
|------------|-------------------------|----------|--|-------------------|--|
| eTraceTpkt | E_TraceLevel
[▶ 449] | Get | I_TpktServerSe
ttingsClass
[▶ 335] | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| | | Set | Local | | |

6.3.2.13 FB_UlosiClientSettingsClass

Der Funktionsbaustein FB_UlosiClientSettingsClass dient der Konfiguration der ULOSI-Kommunikationsschicht (Upper Layer OSI).

Namensraum: Tc3_Ulosi [▶ 97]

Bibliothek: Tc3_Ulosi (Tc3_Ulosi.compiled-library)

Vererbungshierarchie

FB_SocketClientSettingsClass [▶ 200] -> FB_TpktClientSettingsClass [▶ 205] -> FB_Rfc1006ClientSettingsClass [▶ 183] -> FB_UlosiClientSettingsClass

```
FUNCTION_BLOCK FB_UlosiClientSettingsClass EXTENDS FB_Rfc1006ClientSettingsClass IMPLEMENTS I_UlosiClientSettingsClass
```

Schnittstellen

| Typ | Beschreibung |
|------------------------------------|--|
| I_UlosiClientSettingsClass [▶ 335] | Schnittstelle zur Konfiguration der ULOSI-Kommunikationsschicht. |

 **Eigenschaften**

| Name | Typ | Zu-griff | Definitions-ort | Initialwert | Beschreibung |
|---------------------|---|----------|--|---|--|
| bAutoCleanup | BOOL | Get,Set | Geerbt von FB_SocketClientSettingsClass [▶ 200] | TRUE | Aktiviert/Deaktiviert das automatische Aufräumen/Schließen der unbenutzten Sockets (z. B. nach Programmdownload oder Reset der SPS). |
| sSrvNetID | T_AmsNetID [▶ 452] | Get,Set | -/- | “ | TwinCAT Netzwerkadresse des TwinCAT TCP/IP Servers (TF6310). Beim Leerstring wird die Netzwerkadresse des lokalen TwinCAT Systems verwendet. |
| nRemotePort | UDINT | Get,Set | -/- | '127.0.0.1' | TCP/IP Portnummer des Servers. |
| sRemoteHost | T_Ipv4Addr [▶ 452] | Get,Set | -/- | 102 | IPv4-Netzwerkadresse des Servers. |
| tConnect | TIME | Get,Set | -/- | T#45s | Früheste Zeit, nach deren Ablauf eine neue Verbindung aufgebaut werden kann. |
| tAdsTimeout | TIME | Get,Set | -/- | DEFAULT_ADS_TIMEOUT | Maximale Timeoutzeit für ADS-Dienste, sie darf nicht überschritten werden. |
| tSocketTimeout | TIME | Get,Set | -/- | T#5s | Maximale Timeoutzeit die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| bThrottleMode | BOOL | Get,Set | -/- | TRUE | Aktiviert/deaktiviert den Empfangsdaten-Throttle-Polling-Mode. |
| throttleTimes | T_ThrottleTimes [▶ 453] | Get,Set | -/- | Param_Sockets.cDefaultReceiverThrottleTimes | Konfiguration der Zykluszeiten für den Empfangsdaten-Throttle-Polling-Mode. |
| eTraceSocket | E_TraceLevel [▶ 449] | Get,Set | -/- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| ipTLS | I_SocketTlsSettingsClass | Get,Set | -/- | 0 | Konfigurationsparameter der TLS-Protokollschicht (Transport Layer Security). Wert Null = nicht konfiguriert. |
| tTpktTimeout | TIME | Get,Set | Geerbt von FB_TpktClientSettingsClass [▶ 205] | T#10s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceTpkt | E_TraceLevel [▶ 449] | Get,Set | -/- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| sCalling_T_Selector | STRING | Get,Set | Geerbt von FB_Rfc1006ClientSettingsClass [▶ 183] | '00 00' | RFC 1006 Parameter: Calling transport address selector (source). |
| sCalled_T_Selector | STRING | Get,Set | -/- | '00 01' | RFC 1006 Parameter: Called transport address selector (destination). |
| eTpduSize | E_Rfc1006TpduSize [▶ 448] | Get,Set | -/- | E_Rfc1006TpduSize.Size_1024 | Maximale TPDU-Bytelänge. |
| tRfc1006Timeout | TIME | Get,Set | -/- | T#15s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |

| Name | Typ | Zu-griff | Definitions-ort | Initialwert | Beschreibung |
|----------------------|-------------------------|------------|--|-------------------|--|
| eTraceRfc1006 | E_TraceLevel
[▶ 449] | Get,Set | -//- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| sCalling_S_Selector | STRING | Get
Set | I_UlosiClientSettingsClass
[▶ 335]
Local | '00 01' | Calling-S-selector. |
| sCalled_S_Selector | STRING | Get
Set | I_UlosiClientSettingsClass
[▶ 335]
Local | '00 01' | Called-S-selector. |
| nSessionRequirement | WORD | Get
Set | I_UlosiClientSettingsClass
[▶ 335]
Local | 2 | Session requirements. |
| sCalling_P_Selector | STRING | Get
Set | I_UlosiClientSettingsClass
[▶ 335]
Local | '00 00 00 01' | Calling-P-selector. |
| sCalled_P_Selector | STRING | Get
Set | I_UlosiClientSettingsClass
[▶ 335]
Local | '00 00 00 01' | Called-P-selector. |
| nProtocol_Version | BYTE | Get
Set | I_UlosiClientSettingsClass
[▶ 335]
Local | 1 | Protocol version. |
| sContext_Name | STRING | Get
Set | I_UlosiClientSettingsClass
[▶ 335]
Local | '1.0.9506.2.3' | Application context name (object identifier, dotted presentation), MMS. |
| sCalled_AP_Title | STRING | Get
Set | I_UlosiClientSettingsClass
[▶ 335]
Local | '1.1.1.999.1' | Called-AP-title. |
| nCalled_AE_Qualifier | UDINT | Get
Set | I_UlosiClientSettingsClass
[▶ 335]
Local | 12 | Called-AE-qualifier. |
| nCalled_AP_InvID | UDINT | Get
Set | I_UlosiClientSettingsClass
[▶ 335]
Local | 0 | Called-AP-invocation-identifier. |
| nCalled_AE_InvID | UDINT | Get
Set | I_UlosiClientSettingsClass
[▶ 335]
Local | 0 | Called-AE-invocation-identifier. |

| Name | Typ | Zu-griff | Definitions-ort | Initialwert | Beschreibung |
|-----------------------|-------------------------|----------|---|-------------------|--|
| sCalling_AP_Title | STRING | Get | I_UlosiClientSettingsClass
[▶ 335] | '1.1.1.999' | Calling-AP-title. |
| | | Set | Local | | |
| nCalling_AE_Qualifier | UDINT | Get | I_UlosiClientSettingsClass
[▶ 335] | 12 | Calling-AE-qualifier. |
| | | Set | Local | | |
| nCalling_AP_InvID | UDINT | Get | I_UlosiClientSettingsClass
[▶ 335] | 0 | Calling-AP-invocation-identifier. |
| | | Set | Local | | |
| nCalling_AE_InvID | UDINT | Get | I_UlosiClientSettingsClass
[▶ 335] | 0 | Calling-AE-invocation-identifier. |
| | | Set | Local | | |
| nRequirements | BYTE | Get | I_UlosiClientSettingsClass
[▶ 335] | 0 | Requirements (1 == authentication). |
| | | Set | Local | | |
| sMechanism_Name | STRING | Get | I_UlosiClientSettingsClass
[▶ 335] | '2.2.3.1' | Authentication mechanism name. |
| | | Set | Local | | |
| sAuthent_Value | STRING | Get | I_UlosiClientSettingsClass
[▶ 335] | " | Authentication value (password). |
| | | Set | Local | | |
| tUlosiTimeout | TIME | Get | I_UlosiClientSettingsClass
[▶ 335] | T#20s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| | | Set | Local | | |
| eTraceUlosi | E_TraceLevel
[▶ 449] | Get | I_UlosiClientSettingsClass
[▶ 335] | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| | | Set | Local | | |

6.3.2.14 FB_UlosiServerSettingsClass

Der Funktionsbaustein FB_UlosiServerSettingsClass dient der Konfiguration der ULOSI-Kommunikationsschicht (Upper Layer OSI).

Namensraum: [Tc3_Ulosi](#) [▶ 97]

Bibliothek: Tc3_Ulosi (Tc3_Ulosi.compiled-library)

Vererbungshierarchie

[FB_SocketServerSettingsClass](#) [▶ 201] -> [FB_TpktServerSettingsClass](#) [▶ 207] -> [FB_Rfc1006ServerSettingsClass](#) [▶ 186] -> [FB_UlosiServerSettingsClass](#)

```
FUNCTION_BLOCK FB_UlosiServerSettingsClass EXTENDS FB_Rfc1006ServerSettingsClass IMPLEMENTS I_UlosiServerSettingsClass
```

 Schnittstellen

| Typ | Beschreibung |
|--|--|
| I UlosiServerSettingsClass [▶ 336] | Schnittstelle zur Konfiguration der ULOSI-Kommunikationsschicht. |

 **Eigenschaften**

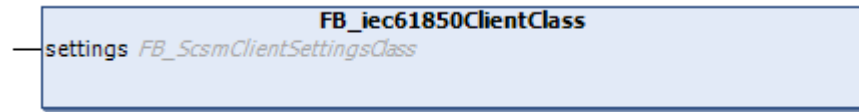
| Name | Typ | Zugriff | Definitionsort | Initialwert | Beschreibung |
|---------------------|--|----------|--|--|---|
| bAutoCleanup | BOOL | Get, Set | Geerbt von FB_SocketServerSettingsClass [▶ 201] | TRUE | Aktiviert/Deaktiviert das automatische Aufräumen/Schließen der unbenutzten Sockets (z. B. nach Programmdownload oder Reset der SPS). |
| bEnable | BOOL | Get, Set | --/ | FALSE | Aktiviert/deaktiviert das Herstellen der Serververbindungen. Beim „TRUE“ wird Listen-Befehl aktiviert (horchen und ankommende Verbindungen werden akzeptiert) und beim „FALSE“ abgelehnt/deaktiviert. |
| sSrvNetID | T_AmsNetID [▶ 452] | Get, Set | --/ | “ | TwinCAT Netzwerkadresse des TwinCAT TCP/IP Servers (TF6310). Beim Leerstring wird die Netzwerkadresse des lokalen TwinCAT Systems verwendet. |
| nLocalPort | UDINT | Get, Set | --/ | '127.0.0.1' | TCP/IP Portnummer des Servers. |
| sLocalHost | T_Ipv4Address [▶ 452] | Get, Set | --/ | 102 | IPv4-Netzwerkadresse des Servers. |
| tAccept | TIME | Get, Set | --/ | T#1s | Wiederholungszykluszeit für die internen TCP/IP-Socket Accept-Befehle. |
| tListen | TIME | Get, Set | --/ | T#1s | Wiederholungszykluszeit für den internen TCP/IP-Socket Listen-Befehl. |
| tAdsTimeout | TIME | Get, Set | --/ | DEFAULT_ADS_TIMEOUT | Maximale Timeoutzeit für ADS-Dienste, sie darf nicht überschritten werden. |
| tSocketTimeout | TIME | Get, Set | --/ | T#5s | Maximale Timeoutzeit die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| bThrottleMode | BOOL | Get, Set | --/ | TRUE | Aktiviert/deaktiviert den Empfangsdaten-Throttle-Polling-Mode. |
| throttleTimes | T_ThrottleTimes [▶ 453] | Get, Set | --/ | Param_Sockets.c DefaultReceiverThrottleTimes | Konfiguration der Zykluszeiten für den Empfangsdaten-Throttle-Polling-Mode. |
| eTraceSocket | E_TraceLevel [▶ 449] | Get, Set | --/ | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| ipTLS | I_SocketTlsSettingsClass [▶ 331] | Get, Set | --/ | 0 | Konfigurationsparameter der TLS-Protokollschicht (Transport Layer Security). Wert Null = nicht konfiguriert. |
| tTpktTimeout | TIME | Get, Set | Geerbt von FB_TpktServerSettingsClass [▶ 207] | T#10s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceTpkt | E_TraceLevel [▶ 449] | Get, Set | --/ | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| sCalling_T_Selector | STRING | Get, Set | Geerbt von FB_Rfc1006ServerSettingsClass [▶ 186] | '00 00' | RFC 1006 Parameter: Calling transport address selector (source). |

| Name | Typ | Zugriff | Definitionsort | Initialwert | Beschreibung |
|----------------------|------------------------------|------------|--|-----------------------------|--|
| sCalled_T_Selector | STRING | Get, Set | -/- | '00 01' | RFC 1006 Parameter: Called transport address selector (destination). |
| eTpduSize | E_Rfc1006TpduSize
[▶ 448] | Get, Set | -/- | E_Rfc1006TpduSize.Size_1024 | Maximale TPDU-Bytelänge. |
| tRfc1006Timeout | TIME | Get, Set | -/- | T#15s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceRfc1006 | E_TraceLevel
[▶ 449] | Get, Set | -/- | E_TraceLevel.None | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| sCalling_S_Selector | STRING | Get
Set | I_UlosiServerSettingsClass
[▶ 336]
Local | '00 01' | Calling-S-selector. |
| sCalled_S_Selector | STRING | Get
Set | I_UlosiServerSettingsClass
[▶ 336]
Local | '00 01' | Called-S-selector. |
| nSessionRequirement | WORD | Get
Set | I_UlosiServerSettingsClass
[▶ 336]
Local | 2 | Session requirements. |
| sCalling_P_Selector | STRING | Get
Set | I_UlosiServerSettingsClass
[▶ 336]
Local | '00 00 00 01' | Calling-P-selector. |
| sCalled_P_Selector | STRING | Get
Set | I_UlosiServerSettingsClass
[▶ 336]
Local | '00 00 00 01' | Called-P-selector. |
| nProtocol_Version | BYTE | Get
Set | I_UlosiServerSettingsClass
[▶ 336]
Local | 1 | Protocol version. |
| sContext_Name | STRING | Get
Set | I_UlosiServerSettingsClass
[▶ 336]
Local | '1.0.9506.2.3' | Application context name (object identifier, dotted presentation), MMS. |
| sCalled_AP_Title | STRING | Get
Set | I_UlosiServerSettingsClass
[▶ 336]
Local | '1.1.1.999.1' | Called-AP-title. |
| nCalled_AE_Qualifier | UDINT | Get
Set | I_UlosiServerSettingsClass
[▶ 336]
Local | 12 | Called-AE-qualifier. |

| Name | Typ | Zu-griff | Definitions-ort | Initialwert | Beschreibung |
|--------------------------------|---------------------------|----------|---|------------------------|--|
| nCalled_A
P_InvID | UDINT | Get | I_UlosiServerSe-
ttingsClass
[▶ 336] | 0 | Called-AP-invocation-identifier. |
| | | Set | Local | | |
| nCalled_A
E_InvID | UDINT | Get | I_UlosiServerSe-
ttingsClass
[▶ 336] | 0 | Called-AE-invocation-identifier. |
| | | Set | Local | | |
| sCalling_
AP_Title | STRING | Get | I_UlosiServerSe-
ttingsClass
[▶ 336] | '1.1.1.999' | Calling-AP-title. |
| | | Set | Local | | |
| nCalling_
AE_Qualif-
ier | UDINT | Get | I_UlosiServerSe-
ttingsClass
[▶ 336] | 12 | Calling-AE-qualifier. |
| | | Set | Local | | |
| nCalling_
AP_InvID | UDINT | Get | I_UlosiServerSe-
ttingsClass
[▶ 336] | 0 | Calling-AP-invocation-identifier. |
| | | Set | Local | | |
| nCalling_
AE_InvID | UDINT | Get | I_UlosiServerSe-
ttingsClass
[▶ 336] | 0 | Calling-AE-invocation-identifier. |
| | | Set | Local | | |
| nRequire-
ments | BYTE | Get | I_UlosiServerSe-
ttingsClass
[▶ 336] | 0 | Requirements (1 == authentication). |
| | | Set | Local | | |
| sMechani-
sm_Name | STRING | Get | I_UlosiServerSe-
ttingsClass
[▶ 336] | '2.2.3.1' | Authentication mechanism name. |
| | | Set | Local | | |
| sAuthent-
Value | STRING | Get | I_UlosiServerSe-
ttingsClass
[▶ 336] | " | Authentication value (password). |
| | | Set | Local | | |
| tUlosiT-
imeout | TIME | Get | I_UlosiServerSe-
ttingsClass
[▶ 336] | T#20s | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| | | Set | Local | | |
| eTraceU-
losi | E_TraceLeve-
l [▶ 449] | Get | I_UlosiServerSe-
ttingsClass
[▶ 336] | E_TraceLevel.Non-
e | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| | | Set | Local | | |

6.3.3 Kommunikation

6.3.3.1 FB_iec61850ClientClass



Der Funktionsbaustein FB_iec61850ClientClass implementiert Schnittstellen zum Kommunikationsaufbau und Datenaustausch zwischen der TwinCAT SPS Client-Applikation und dem IEC 61850 Server.

Namensraum: [Tc3_iec61850](#) [[► 97](#)]

Bibliothek: Tc3_iec61850 (Tc3_iec61850.compiled-library)

Vererbungshierarchie

FB_iec61850ClientClass

```

FUNCTION_BLOCK FB_iec61850ClientClass IMPLEMENTS
    I_ScsmClientClass,
    I_ScsmSystemClockEventSink,
    I_ScsmAbortIndEventSink,
    I_ScsmAssociateCnfEventSink,
    I_ScsmReleaseIndEventSink,
    I_ScsmReleaseCnfEventSink,
    I_ScsmGetDataValuesCnfEventSink,
    I_ScsmSetDataValuesCnfEventSink,
    I_ScsmControlCnfEventSink,
    I_ScsmGetServerDirectoryCnfEventSink,
    I_ScsmGetLogicalDeviceDirectoryCnfEventSink,
    I_ScsmGetLogicalNodeDirectoryCnfEventSink,
    I_ScsmCreateDataSetCnfEventSink,
    I_ScsmDeleteDataSetCnfEventSink,
    I_ScsmGetDataSetDirectoryCnfEventSink,
    I_ScsmReportIndEventSink,
    I_ScsmLastApplErrorIndEventSink,
    I_ScsmCommandTerminationIndEventSink,
    I_ScsmIdentifyCnfEventSink,
    I_ScsmStatusCnfEventSink
VAR_INPUT
    settings : FB_ScsmClientSettingsClass;
END_VAR
  
```


 **Schnittstellen**

| Typ | Beschreibung |
|---|---|
| I_ScsmClientClass [▶ 351] | Schnittstellen zum Kommunikationsaufbau und Datenaustausch. |
| I_ScsmAbortIndEventSink [▶ 350] | Schnittstelle der Abort-Dienst Ereignisbehandlungsroutine. |
| I_ScsmAssociateCnfEventSink [▶ 350] | Schnittstelle der Associate-Dienst Ereignisbehandlungsroutine. |
| I_ScsmReleaseCnfEventSink [▶ 383] | Schnittstelle der Release-Dienst Ereignisbehandlungsroutine. |
| I_ScsmReleaseIndEventSink [▶ 383] | Schnittstelle der Release-Dienst Ereignisbehandlungsroutine. |
| I_ScsmGetDataValuesCnfEventSink [▶ 379] | Schnittstelle der GetDataValues-Dienst Ereignisbehandlungsroutine. |
| I_ScsmSetDataValuesCnfEventSink [▶ 385] | Schnittstelle der SetDataValues Ereignisbehandlungsroutine. |
| I_ScsmControlCnfEventSink [▶ 375] | Schnittstelle der Control-Dienst Ereignisbehandlungsroutine für die Schaltersteuerung (Operate, Cancel, Select, SelectWithValue). |
| I_ScsmGetServerDirectoryCnfEventSink [▶ 381] | Schnittstelle der GetServerDirectory-Dienst Ereignisbehandlungsroutine. |
| I_ScsmGetLogicalDeviceDirectoryCnfEventSink [▶ 379] | Schnittstelle der GetLogicalDeviceDirectory-Dienst Ereignisbehandlungsroutine. |
| I_ScsmGetLogicalNodeDirectoryCnfEventSink [▶ 380] | Schnittstelle der GetLogicalNodeDirectory-Dienst Ereignisbehandlungsroutine. |
| I_ScsmCreateDataSetCnfEventSink [▶ 377] | Schnittstelle der CreateDataSet-Dienst Ereignisbehandlungsroutine. |
| I_ScsmDeleteDataSetCnfEventSink [▶ 377] | Schnittstelle der DeleteDataSet-Dienst Ereignisbehandlungsroutine. |
| I_ScsmGetDataSetDirectoryCnfEventSink [▶ 378] | Schnittstelle der GetDataSetDirectory-Dienst Ereignisbehandlungsroutine. |
| I_ScsmSystemClockEventSink [▶ 386] | Schnittstelle der SystemClock-Dienst Ereignisbehandlungsroutine. |
| I_ScsmReportIndEventSink [▶ 384] | Schnittstelle der Report-Dienst Ereignisbehandlungsroutine. |
| I_ScsmLastApplErrorIndEventSink [▶ 382] | Schnittstelle der LastApplError-Dienst Ereignisbehandlungsroutine. |
| I_ScsmCommandTerminationIndEventSink [▶ 376] | Schnittstelle der Befehlsbeendigung-Ereignisbehandlungsroutine für Schaltersteuerung. |
| I_ScsmIdentifyCnfEventSink [▶ 382] | Schnittstelle der MMS-Identify-Dienst Ereignisbehandlungsroutine. |
| I_ScsmStatusCnfEventSink [▶ 386] | Schnittstelle der MMS-Status-Dienst Ereignisbehandlungsroutine. |

 **Eingänge**

| Name | Typ | Beschreibung |
|----------|--|---|
| settings | FB_ScsmClientSettingsClass [▶ 189] | Protokolleinstellungen der Kommunikationsschicht. |

 **Methoden**

| Name | Beschreibung |
|--|---|
| Execute | Diese Methode triggert die Ausführung der Kommunikationsschicht. |
| AssociateReq [▶ 355] | Aktiviert den Dienst: Associate. |
| ReleaseReq [▶ 356] | Aktiviert den Dienst: Release. |
| ReleaseRsp [▶ 356] | Beantwortet den Dienst: Release. |
| AbortReq [▶ 357] | Aktiviert den Dienst: Abort. |
| ControlCancelReq [▶ 357] | Aktiviert den Dienst: Cancel. |
| ControlOperateReq [▶ 358] | Aktiviert den Dienst: Operate. |
| ControlSelectReq [▶ 359] | Aktiviert den Dienst: Select. |
| ControlSelectWithValueReq [▶ 360] | Aktiviert den Dienst: SelectWithValue. |
| CreateDataSetReq [▶ 361] | Aktiviert den Dienst: CreateDataSet. |
| DeleteDataSetReq [▶ 362] | Aktiviert den Dienst: DeleteDataSet. |
| GetAllDataValuesReq [▶ 363] | Aktiviert den Dienst: GetAllDataValues. |
| GetAllLogicalDeviceValuesReq [▶ 363] | Aktiviert den Dienst: GetAllLogicalDeviceValues (spezifisch TwinCAT). |
| GetAllServerValuesReq [▶ 364] | Aktiviert den Dienst: GetAllServerValues (spezifisch TwinCAT). |
| GetBrCBValuesReq [▶ 364] | Aktiviert den Dienst: GetBRCBValues. |
| GetDataSetDirectoryReq [▶ 365] | Aktiviert den Dienst: GetDataSetDirectory. |
| GetDataSetValuesReq [▶ 366] | Aktiviert den Dienst: GetDataSetValues. |
| GetDataValuesReq [▶ 366] | Aktiviert den Dienst: GetDataValues. |
| GetGoCBValuesReq [▶ 367] | Aktiviert den Dienst: GetGoCBValues. |
| GetGsCBValuesReq | Aktiviert den Dienst: GetGsCBValues. |
| GetLCBValuesReq | Aktiviert den Dienst: GetLCBValues. |
| GetLogicalDeviceDirectoryReq [▶ 368] | Aktiviert den Dienst: GetLogicalDeviceDirectory. |
| GetLogicalNodeDirectoryReq [▶ 368] | Aktiviert den Dienst: GetLogicalNodeDirectory. |
| GetMsvCBValuesReq | Aktiviert den Dienst: GetMSVCBValues. |
| GetServerDirectoryReq [▶ 369] | Aktiviert den Dienst: GetServerDirectory. |
| GetSgCBValuesReq | Aktiviert den Dienst: GetSGCBValuesReq. |
| GetUrCBValuesReq [▶ 370] | Aktiviert den Dienst: GetURCBValues. |
| GetUsvCBValuesReq | Aktiviert den Dienst: GetUSVCBValues. |
| SetAllDataValuesReq [▶ 370] | Aktiviert den Dienst: SetAllDataValues (spezifisch TwinCAT). |
| SetBrCBValuesReq [▶ 371] | Aktiviert den Dienst: SetBRCBValues. |
| SetDataSetValuesReq [▶ 371] | Aktiviert den Dienst: SetDataSetValues (spezifisch TwinCAT). |
| SetDataValuesReq [▶ 372] | Aktiviert den Dienst: SetDataValues. |
| SetGoCBValuesReq [▶ 373] | Aktiviert den Dienst: SetGoCBValues. |
| SetGsCBValuesReq | Aktiviert den Dienst: SetGsCBValues. |
| SetLCBValuesReq | Aktiviert den Dienst: SetLCBValues. |
| SetMsvCBValuesReq | Aktiviert den Dienst: SetMSVCBValues. |
| SetSgCBValuesReq | Aktiviert den Dienst: SetSGCBValues. |
| SetUrCBValuesReq [▶ 373] | Aktiviert den Dienst: SetURCBValues. |
| SetUsvCBValuesReq | Aktiviert den Dienst: SetUSVCBValues. |
| IdentifyReq [▶ 374] | Aktiviert den Dienst: MMS-Identify. |
| StatusReq [▶ 375] | Aktiviert den Dienst: MMS-Status. |

 **Methoden zur Ereignisbehandlung (Callback-Methoden)**

| Name | Beschreibung |
|--|--|
| OnAssociateCnf [▶ 351] | Benutzerdefinierte Associate-Dienst Ereignisbehandlungsroutine. |
| OnReleaseCnf [▶ 383] | Benutzerdefinierte Release-Dienst Ereignisbehandlungsroutine. |
| OnReleaseInd [▶ 384] | Benutzerdefinierte Release-Dienst Ereignisbehandlungsroutine. |
| OnAbortInd [▶ 350] | Benutzerdefinierte Abort-Dienst Ereignisbehandlungsroutine. |
| OnControlCnf [▶ 376] | Benutzerdefinierte Control-Dienst Ereignisbehandlungsroutine für die Schaltersteuerung (Operate, Cancel, Select, SelectWithValue). |
| OnCreateDataSetCnf [▶ 377] | Benutzerdefinierte CreateDataSet-Dienst Ereignisbehandlungsroutine. |
| OnDeleteDataSetCnf [▶ 378] | Benutzerdefinierte DeleteDataSet-Dienst Ereignisbehandlungsroutine. |
| OnGetDataSetDirectoryCnf [▶ 378] | Benutzerdefinierte GetDataSetDirectory-Dienst Ereignisbehandlungsroutine. |
| OnGetDataValuesCnf [▶ 379] | Benutzerdefinierte GetDataValues-Dienst Ereignisbehandlungsroutine. |
| OnGetLogicalDeviceDirectoryCnf [▶ 380] | Benutzerdefinierte GetLogicalDeviceDirectory-Dienst Ereignisbehandlungsroutine. |
| OnGetLogicalNodeDirectoryCnf [▶ 380] | Benutzerdefinierte GetLogicalNodeDirectory-Dienst Ereignisbehandlungsroutine. |
| OnGetServerDirectoryCnf [▶ 381] | Benutzerdefinierte GetServerDirectory-Dienst Ereignisbehandlungsroutine. |
| OnLastApplErrorInd [▶ 383] | Benutzerdefinierte LastApplError-Dienst Ereignisbehandlungsroutine. |
| OnBufferedReportInd [▶ 384] | Benutzerdefinierte Buffered-Report-Dienst Ereignisbehandlungsroutine. |
| OnUnbufferedReportInd [▶ 385] | Benutzerdefinierte Unbuffered-Report-Dienst Ereignisbehandlungsroutine. |
| OnSetDataValuesCnf [▶ 385] | Benutzerdefinierte SetDataValues-Dienst Ereignisbehandlungsroutine. |
| OnGetSystemTime [▶ 387] | Benutzerdefinierte SystemClock-Dienst Ereignisbehandlungsroutine. |
| OnCommandTerminationInd [▶ 376] | Benutzerdefinierte Befehlsbeendigung-Ereignisbehandlungsroutine für Schaltersteuerung. |
| OnIdentifyCnf [▶ 382] | Benutzerdefinierte MMS-Identify-Dienst Ereignisbehandlungsroutine. |
| OnStatusCnf [▶ 386] | Benutzerdefinierte MMS-Status-Dienst Ereignisbehandlungsroutine. |

 **Eigenschaften**

| Name | Typ | Zu-griff | Beschreibung |
|---|--|------------|---|
| ipIED [▶ 375] | I_AcsiCommonIntelligentElectronicDeviceClass [▶ 316] | Get
Set | Schnittstellenzeiger von einem Objekt mit der Implementierung der obersten Ebene des IEC 61850 Datenmodells. |
| eState | E_AsyncEnvironmentState [▶ 443] | Get | Umgebungszustand der Kommunikationsverbindung. |
| ipSettings | I_ScsmClientSettingsClass [▶ 328] | Get | Protokolleinstellungen der Kommunikationsschicht. |
| ipAssociateCnf | I_ScsmAssociateCnfEventSink [▶ 350] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnAssociateCnf [▶ 351]. |
| ipReleaseCnf | I_ScsmReleaseCnfEventSink [▶ 383] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnReleaseCnf [▶ 383]. |
| ipReleaseInd | I_ScsmReleaseIndEventSink [▶ 383] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnReleaseInd [▶ 384]. |
| ipAbortInd | I_ScsmAbortIndEventSink [▶ 350] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnAbortInd [▶ 350]. |
| ipSystemClock | I_ScsmSystemClockEventSink [▶ 386] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnGetSystemTime [▶ 387]. |
| ipControlCnf | I_ScsmControlCnfEventSink [▶ 375] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnControlCnf [▶ 376]. |
| ipCreateDataSetCnf | I_ScsmCreateDataSetCnfEventSink [▶ 377] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnCreateDataSetCnf [▶ 377]. |
| ipDeleteDataSetCnf | I_ScsmDeleteDataSetCnfEventSink [▶ 377] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnDeleteDataSetCnf [▶ 378]. |
| ipGetDataSetDirectory | I_ScsmGetDataSetDirectoryEventSink [▶ 378] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnGetDataSetDirectoryCnf [▶ 378]. |
| ipGetDataValuesCnf | I_ScsmGetDataValuesCnfEventSink [▶ 379] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnGetDataValuesCnf [▶ 379]. |
| ipGetLogicalDeviceDirectoryCnf | I_ScsmGetLogicalDeviceDirectoryCnfEventSink [▶ 379] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnGetLogicalDeviceDirectoryCnf [▶ 380]. |

| Name | Typ | Zu-griff | Beschreibung |
|------------------------------|---|----------|---|
| ipGetLogicalNodeDirectoryCnf | I_ScsmGetLogicalNodeDirectoryCnfEventSink [▶ 380] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnGetLogicalNodeDirectoryCnf [▶ 380]. |
| ipGetServerDirectoryCnf | I_ScsmGetServerDirectoryCnfEventSink [▶ 381] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnGetServerDirectoryCnf [▶ 381]. |
| ipIdentifyCnf | I_ScsmIdentifyCnfEventSink [▶ 382] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnIdentifyCnf [▶ 382]. |
| ipLastAppLErrorInd | I_ScsmLastAppLErrorIndEventSink [▶ 382] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnLastAppLErrorInd [▶ 383]. |
| ipReportInd | I_ScsmReportIndEventSink [▶ 384] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutinen: OnBufferedReportInd [▶ 384] und OnUnbufferedReportInd [▶ 385]. |
| ipSetDataValuesCnf | I_ScsmSetDataValuesCnfEventSink [▶ 385] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnSetDataValuesCnf [▶ 385]. |
| ipStatusCnf | I_ScsmStatusCnfEventSink [▶ 386] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnStatusCnf [▶ 386]. |
| ipCommandTerminationInd | I_ScsmCommandTerminationIndEventSink [▶ 376] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnCommandTerminationInd [▶ 376]. |

6.3.3.2 FB_iec61850ServerClass

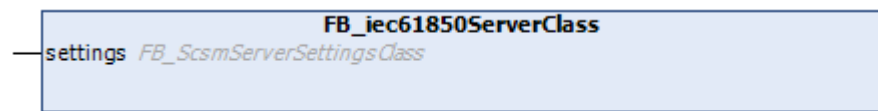


Abb. 1:

Der Funktionsbaustein FB_iec61850ServerClass implementiert eine Schnittstelle zum Kommunikationsaufbau und Datenaustausch zwischen der TwinCAT Server-Applikation und dem IEC 61850 Client.

Namensraum: [Tc3_iec61850](#) [▶ 97]

Bibliothek: Tc3_iec61850 (Tc3_iec61850.compiled-library)

Vererbungshierarchie

FB_iec61850ServerClass

```
FUNCTION_BLOCK FB_iec61850ServerClass IMPLEMENTS I_ScsmServerClass
VAR_INPUT
    settings : FB_ScsmServerSettingsClass;
END_VAR
```

 **Schnittstellen**

| Typ | Beschreibung |
|-------------------|---|
| I_ScsmServerClass | Schnittstellen zum Kommunikationsaufbau und Datenaustausch. |

 **Eingänge**

| Name | Typ | Beschreibung |
|----------|--|---|
| settings | FB_ScsmServerSettingsClass [▶ 195] | Protokolleinstellungen der Kommunikationsschicht. |

 **Methoden**

| Name | Beschreibung |
|------------------|---|
| AddConnection | Fügt eine Verbindungsinstanz zum Server hinzu. |
| Execute | Diese Methode löst die Ausführung der Kommunikationsschicht sowie der Verbindungsinstanzen aus. |
| RemoveConnection | Entfernt eine Verbindungsinstanz vom Server. |

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|---------------|--|---------|---|
| ipConnections | I_ScsmConnectionClass | Get | Schnittstellenzeiger von einem Objekt mit der Implementierung der ersten Verbindungsinstanz. |
| ipIED | I_AcsiCommonIntelligentElectronicDeviceClass [▶ 316] | Get | Schnittstellenzeiger von einem Objekt mit der Implementierung der obersten Ebene des IEC 61850 Datenmodells. |
| | | Set | |
| ipSettings | I_ScsmServerSettingsClass [▶ 329] | Get | Protokolleinstellungen der Kommunikationsschicht. |
| | | Set | |
| ipSystemClock | I_ScsmSystemClockEventSink [▶ 386] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnGetSystemTime [▶ 387] . |
| nItems | UDINT | Get | Die Anzahl an Verbindungsinstanzen. |
| sDesc | STRING | Get | Die Beschreibung des Servers. |
| | | Set | |

6.3.3.3 Steuerblock-Zugriff

6.3.3.3.1 FB_ScsmBrCBImplClass

Der Funktionsbaustein FB_ScsmBrCBImplClass beinhaltet die Client- und Server-Implementierung der Eigenschaften und Funktionen eines BRCB-Steuerblocks. Für jeden konfigurierten GOOSE-Steuerblock wird vom TwinCAT Telecontrol Configurator eine Instanz von FB_GseBrCBImplClass angelegt.

Namensraum: [Tc3_iec61850_8_1 \[▶ 97\]](#)

Bibliothek: Tc3_iec61850_8_1 (Tc3_iec61850_8_1.compiled-library)

Vererbungshierarchie

[FB_AcsiCommonNodeClass \[▶ 154\]](#) -> [FB_AcsiCommonDataClass \[▶ 148\]](#) -
> [FB_AcsiCommonControlBlockClass \[▶ 147\]](#) -> [FB_AcsiCommonBufferedReportControlBlockClass \[▶ 147\]](#) -
> [FB_AcsiBufferedReportControlBlock \[▶ 164\]](#) -> [FB_ScsmBrCBImplClass](#)

```
FUNCTION_BLOCK FB_ScsmBrCBImplClass EXTENDS FB_AcsiBufferedReportControlBlock IMPLEMENTS I_ScsmBrCBImplClass
VAR_INPUT
    Client : FB_ScsmBrCBClientJobClass;
    Server : FB_ScsmBrCBServerJobClass;
END_VAR
```

Schnittstellen

| Typ | Beschreibung |
|---|---|
| I_ScsmBrCBImplClass [▶ 342] | Schnittstelle der BRCB-Steuerblock Implementierung. |

 **Eingänge**

| Name | Typ | Definitionsort | Beschreibung |
|-------------|--|--|---|
| Client | FB_ScsmBrCBClientJobClass [230] | Local | Client-Implementierung des BRCB-Steuerblocks. |
| Server | FB_ScsmBrCBServerJobClass [231] | Local | Server-Implementierung des BRCB-Steuerblocks. |
| RptID | FB_AcsiDATypeVisString129 [144] | Geerbt von FB_AcsiBufferedReportControlBlock [164] | Datenattribut, Ed1=>RptID:VisStr[65]! |
| RptEna | FB_AcsiBATypeBOOLEAN [98] | -//- | Datenattribut |
| DatSet | FB_AcsiDATypeDatSetRef [146] | -//- | Datenattribut |
| ConfRev | FB_AcsiBATypeINT32U [101] | -//- | Datenattribut |
| OptFlds | FB_AcsiDATypeOptFlds [123] | -//- | Datenattribut |
| BufTm | FB_AcsiBATypeINT32U [101] | -//- | Datenattribut |
| SqNum | FB_AcsiBATypeINT16U [100] | -//- | Datenattribut |
| TrgOps | FB_AcsiDATypeTriggerConditions [141] | -//- | Datenattribut |
| IntgPd | FB_AcsiBATypeINT32U [101] | -//- | Datenattribut |
| GI | FB_AcsiBATypeBOOLEAN [98] | -//- | Datenattribut |
| PurgeBuf | FB_AcsiBATypeBOOLEAN [98] | -//- | Datenattribut |
| EntryID | FB_AcsiDATypeEntryID [114] | -//- | Datenattribut |
| TimeOfEntry | FB_AcsiDATypeEntryTime [114] | -//- | Datenattribut |
| ResvTms | FB_AcsiBATypeINT16 [100] | -//- | Datenattribut, Ed2. |
| Owner | FB_AcsiDATypeOctet64 [122] | -//- | Datenattribut, Ed2. |

 **Eigenschaften**

| Name | Type | Zugriff | Definitionsort | Beschreibung |
|------------------|--|-------------|---|---|
| bGI | BOOL | Get,
Set | I_ScsmBrCBImpl
Class [▶ 342] | Attributwert: Aktivierung des Generalabfrage-Befehls. |
| bPurge
Buf | BOOL | Get,
Set | -//- | Attributwert: Aktivierung des Purge-Befehls. |
| bRptEn
a | BOOL | Get,
Set | -//- | Attributwert: Aktivierung/Deaktivierung des RCB-Steuerblocks. Weitere Konfiguration benötigt. |
| cOptFld
s | ST_AcsiOptionalFields [▶ 459] | Get,
Set | -//- | Attributwert: Konfiguration der optionalen Felder, die mit einem Report übertragen werden. |
| cTrgOp
s | ST_AcsiTriggerConditions [▶ 466] | Get,
Set | -//- | Attributwert: Konfiguration der Auslösoptionen für das Senden eines Reports. |
| iResvT
ms | INT | Get,
Set | -//- | Attributwert: Konfiguration der Zeit für die Reservierung des RCBs für einen bestimmten Client. |
| nBufTm | DWORD | Get,
Set | -//- | Attributwert: Konfiguration der max. Report-Pufferzeit in Millisekunden. |
| nConfR
ev | DWORD | Get,
Set | -//- | Attributwert: Revisionsnummer der Konfiguration des RCB-Steuerblocks. |
| nEntryI
D | LWORD | Get,
Set | -//- | Attributwert: ID des Report-Eintrags als 64-Bit Zahlenwert. |
| nIntgPd | DWORD | Get,
Set | -//- | Attributwert: Konfiguration der max. Zeit zwischen den Integrity-Reports in Millisekunden. |
| nSqNu
m | WORD | Get,
Set | -//- | Attributwert: Report-Sequenznummer. |
| oEntryI
D | T_OCTET8 [▶ 478] | Get,
Set | -//- | Attributwert: ID des Report-Eintrags als Oktett-String. |
| oOwner | T_OCTET64 [▶ 479] | Get,
Set | -//- | Attributwert: Eigentümer des RCBs. IP-Adresse des Clients welcher den RCB für sich reserviert hat (Offline-/Online-Reservierung). |
| sDatSet | T_AcsiObjectReference [▶ 476] | Get,
Set | -//- | Attributwert: Name des verknüpften Data-Sets. |
| sRptID | T_AcsiVisString129 [▶ 476] | Get,
Set | -//- | Attributwert: Report-Identifikationsstring. |
| tTimeOf
Entry | T_BinaryTime [▶ 476] | Get,
Set | -//- | Attributwert: Report-Zeitstempel. |
| ipDatSe
t | I_AcsiCommonDataSetClass [▶ 313] | Get | -//- | Schnittstellenzeiger des verknüpften Data-Sets. |
| ipClient | I_ScsmBrCBClientJobClass [▶ 343] | Get | -//- | Schnittstellenzeiger der Client-Implementierung des BRCB-Steuerblocks. |
| ipServer | I_ScsmBrCBServerJobClass [▶ 345] | Get | -//- | Schnittstellenzeiger der Server-Implementierung des BRCB-Steuerblocks. |

6.3.3.3.2 FB_ScsmBrCBClientJobClass

Client-Implementierung des BRCB-Steuerblocks.

Namensraum: [Tc3_iec61850_8_1 \[▶ 97\]](#)

Bibliothek: [Tc3_iec61850_8_1 \(Tc3_iec61850_8_1.compiled-library\)](#)

Vererbungshierarchie

[FB_AsyncServiceResultClass \[▶ 244\]](#) -> [FB_ScsmBrCBClientJobClass](#)

```
FUNCTION_BLOCK FB_ScsmBrCBClientJobClass EXTENDS FB_AsyncServiceResultClass IMPLEMENTS I_ScsmBrCBClientJobClass
VAR_OUTPUT
    stInfo : ST_ScsmBrCBJobInfo;
END_VAR
```

 **Schnittstellen**

| Typ | Beschreibung |
|--|--|
| I_ScsmBrCBClientJobClass [▶ 343] | Client-Schnittstelle der BRCB-Steuerblock Implementierung. |

 **Ausgänge**

| Name | Typ | Beschreibung |
|--------|--|---|
| stInfo | ST_ScsmBrCBJobInfo [▶ 471] | Information, die in der letzten Report Nachricht empfangen wurde. |

 **Methoden**

| Name | Beschreibung |
|---------------------------------------|--------------------------------------|
| EnableReq [▶ 344] | Aktiviert den Steuerblock. |
| DisableReq [▶ 344] | Deaktiviert den Steuerblock. |
| Glreq [▶ 344] | Aktiviert den Generalabfrage-Befehl. |
| PurgeBufReq [▶ 345] | Aktiviert den Purge-Befehl. |
| ResyncReq [▶ 345] | Aktiviert den Resync-Befehl. |

 **Eigenschaften**

| Name | Typ | Zu-griff | Beschreibung |
|----------|--|----------|--|
| cOptFlds | ST_AcsiOptionalFields [▶ 459] | Set | Optionalen Felder, die in einem Report übertragen werden. |
| cTrgOps | ST_AcsiTriggerConditions [▶ 466] | Set | Auslösoptionen für das Senden eines Reports. |
| iResvTms | INT(-1..3600) | Set | Konfiguriert die Zeit für die Reservierung des RCBs für einen bestimmten Client. |
| nBufTm | DWORD | Set | Max. Report-Pufferzeit in Millisekunden. |
| nIntgPd | DWORD | Set | Max. Zeit zwischen den Integrity-Reports in Millisekunden. |
| sRptID | T_AcsiVisString129 [▶ 476] | Set | Report-Identifikationsstring. |

6.3.3.3 FB_ScsmBrCBServerJobClass

Server-Implementierung des BRCB-Steuerblocks.

Namensraum: Tc3_iec61850_8_1 [[▶ 97](#)]

Bibliothek: Tc3_iec61850_8_1 (Tc3_iec61850_8_1.compiled-library)

Vererbungshierarchie

FB_AsyncServiceResultClass [[▶ 244](#)] -> FB_ScsmBrCBServerJobClass

```

FUNCTION_BLOCK FB_ScsmBrCBServerJobClass EXTENDS FB_AsyncServiceResultClass IMPLEMENTS I_ScsmBrCBServerJobClass
VAR_OUTPUT
    stInfo : ST_ScsmBrCBJobInfo;
END_VAR

```

Schnittstellen

| Typ | Beschreibung |
|--|--|
| I_ScsmBrCBServerJobClass [▶ 345] | Server-Schnittstelle der BRCB-Steuerblock Implementierung. |

Ausgänge

| Name | Typ | Beschreibung |
|--------|--|--|
| stInfo | ST_ScsmBrCBJobInfo [▶ 471] | Information, die in der letzten Report Nachricht gesendet wurde. |

Methoden

| Name | Beschreibung |
|---|---|
| Execute [▶ 345] | Führt die RCB-Zustandsmaschine des Servers aus. |

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|----------------------------|---------|---------|--|
| bOverwriteSameCycleChanges | BOOLEAN | Set | Bei TRUE werden mehrfache Wertänderungen innerhalb eines SPS-Zyklus und eines Datenobjekts als einzelne Wertänderungen behandelt (es wird nur ein Report generiert). Bei FALSE generiert jede Wertänderung einen eigenen Report. |
| bReplaceBufferedMX | BOOLEAN | Set | Konfiguriert, ob Daten der funktionalen Gruppe „MX“, gesondert, wie Daten der funktionalen Gruppe „ST“, behandelt werden sollen. |
| nMaxBufferEntries | UDINT | Set | Konfiguriert die max. Anzahl der gepufferten Report-Einträgen. |
| sResvClient | STRING | Set | String mit der IP-Adresse des in SCL vorkonfigurierten Clients (Offline-Reservierung). Beispiel: '192.168.10.140'. Wenn gesetzt, dann wird im Server der Zugriff auf die RCB-Instanz auf den Client mit dieser IP-Adresse beschränkt („ClientLN“-Einträge in der SCL). Default: Leerstring (Online-Reservierung). Beim Leerstring kann jeder Client, dynamisch den Zugriff auf den RCB für sich reservieren. |

6.3.3.3.4 FB_ScsmGoCBImplClass

Der Funktionsbaustein FB_ScsmGoCBImplClass beinhaltet die Client- und Server-Implementierung der Eigenschaften und Funktionen eines GOOSE-Steuerblocks.

Namensraum: [Tc3_iec61850_8_1](#) [[▶ 97](#)]

Bibliothek: [Tc3_iec61850_8_1](#) ([Tc3_iec61850_8_1.compiled-library](#))

Vererbungshierarchie

[FB_AcsiCommonNodeClass](#) [[▶ 154](#)] -> [FB_AcsiCommonDataClass](#) [[▶ 148](#)] -

> [FB_AcsiCommonControlBlockClass](#) [[▶ 147](#)] -> [FB_AcsiCommonGooseControlBlockClass](#) [[▶ 150](#)] -

> [FB_AcsiGooseControlBlock](#) [[▶ 165](#)] -> [FB_ScsmGoCBImplClass](#)

```
FUNCTION_BLOCK FB_ScsmGoCBImplClass EXTENDS FB_AcsiGooseControlBlock IMPLEMENTS I_ScsmGoCBImplClass
VAR_INPUT
    Client : FB_ScsmGoCBClientJobClass;
    Server : FB_ScsmGoCBServerJobClass;
END_VAR
```

 **Schnittstellen**

| Typ | Beschreibung |
|---|--|
| I_ScsmGoCBImplClass [▶ 346] | Schnittstelle der GOOSE-Steuerblock Implementierung. |

 **Eingänge**

| Name | Typ | Definitionsort | Beschreibung |
|------------|---|---|--|
| Client | FB_ScsmGoCBClientJobClass [▶ 234] | Local | Client-Implementierung des GOOSE-Steuerblocks. |
| Server | FB_ScsmGoCBServerJobClass [▶ 235] | Local | Server-Implementierung des GOOSE-Steuerblocks. |
| GoEna | FB_AcsiBATypeBOOLEAN [▶ 98] | Geerbt von FB_AcsiGooseControlBlock [▶ 165] | Datenattribut |
| GoID | FB_AcsiDATypeVisString129 [▶ 144] | -/- | Datenattribut |
| DatSet | FB_AcsiDATypeDataSetRef [▶ 146] | -/- | Datenattribut |
| ConfRev | FB_AcsiBATypeINT32U [▶ 101] | -/- | Datenattribut |
| NdsCom | FB_AcsiBATypeBOOLEAN [▶ 98] | -/- | Datenattribut |
| DstAddress | FB_AcsiDATypePhyComAddr [▶ 127] | -/- | Datenattribut |
| MinTime | FB_AcsiBATypeINT32U [▶ 101] | -/- | Datenattribut |
| MaxTime | FB_AcsiBATypeINT32U [▶ 101] | -/- | Datenattribut |
| FixedOffs | FB_AcsiBATypeBOOLEAN [▶ 98] | -/- | Datenattribut |

 **Eigenschaften**

| Name | Type | Zugriff | Definitionsort | Beschreibung |
|------------|--|----------|---|--|
| bFixedOffs | BOOL | Get | I_ScsmGoCBImplClass [▶ 346] | Attributwert: Aktiviert/Deaktiviert Kodierung mit festen Längen. |
| bGoEna | BOOL | Get, Set | -//- | Attributwert: Aktiviert/Deaktiviert den GOOSE-Steuerblock. |
| bNdsCom | BOOL | Get | -//- | Attributwert: Gibt an, ob der GOOSE-Steuerblock weitere Konfiguration benötigt. |
| ePRIORITY | E_AcsiVlanPriority [▶ 442] | Get | -//- | Prioritätsstufe im VLAN-Netz. |
| ipDatSet | I_AcsiCommonDataSetClass [▶ 313] | Get | -//- | Schnittstellenzeiger auf das verknüpfte Data-Set. |
| macAddr | ETHERNET_ADDRESS [▶ 451] | Get | -//- | Ziel-MAC-Adresse. |
| nAPPID | WORD | Get | -//- | Application Identifier. |
| nConfRev | UDINT | Get | -//- | Revisionsnummer der Konfiguration des GOOSE-Steuerblocks. |
| nMaxTime | DWORD | Get | -//- | Maximale Zeit zwischen den wiederholten Übertragungen einer GOOSE-Nachricht in Millisekunden. |
| nMinTime | DWORD | Get | -//- | Zeit zwischen dem direkten Senden nach Datenänderung und der ersten Wiederholung in Millisekunden. |
| nVID | WORD(0..4095) | Get | -//- | ID des VLAN-Netzes. |
| oAddr | T_OCTET6 [▶ 478] | Get | -//- | Ziel-MAC-Adresse. |
| sAddr | STRING(17) | Get | -//- | Ziel-MAC-Adresse. |
| sDatSet | T_AcsiObjectReference [▶ 476] | Get, Set | -//- | Name des verknüpften Data-Sets. |
| sGoID | T_AcsiVisString129 [▶ 476] | Get, Set | -//- | Die ID der GOOSE-Nachrichten. |
| ipClient | I_ScsmGoCBClientJobClass [▶ 346] | Get | -//- | Schnittstellenzeiger der Client-Implementierung des GOCB-Steuerblocks. |
| ipServer | I_ScsmGoCBServerJobClass [▶ 347] | Get | -//- | Schnittstellenzeiger der Server-Implementierung des GOCB-Steuerblocks. |

6.3.3.3.5 FB_ScsmGoCBClientJobClass

Client-Implementierung des GOCB-Steuerblocks.

Namensraum: [Tc3_iec61850_8_1](#) [[▶ 97](#)]

Bibliothek: [Tc3_iec61850_8_1](#) ([Tc3_iec61850_8_1.compiled-library](#))

Vererbungshierarchie

[FB_AsyncServiceResultClass](#) [[▶ 244](#)] -> [FB_ScsmGoCBClientJobClass](#)

```
FUNCTION_BLOCK FB_ScsmGoCBClientJobClass EXTENDS FB_AsyncServiceResultClass IMPLEMENTS I_ScsmGoCBClientJobClass
VAR_OUTPUT
    stInfo : ST_ScsmGoCBJobInfo;
END_VAR
```

 Schnittstellen

| Typ | Beschreibung |
|--|--|
| I_ScsmGoCBClientJobClass [▶ 346] | Client-Schnittstelle der GOCB-Steuerblock Implementierung. |

 Ausgänge

| Name | Typ | Beschreibung |
|---------|--|--------------|
| stlInfo | ST_ScsmGoCBJobInfo [▶ 473] | |

 Methoden

| Name | Beschreibung |
|------------------------------------|------------------------------|
| EnableReq [▶ 347] | Aktiviert den Steuerblock. |
| DisableReq [▶ 347] | Deaktiviert den Steuerblock. |

 Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|-------|--|---------|--------------------------------------|
| sGoID | T_AcsiVisString129 [▶ 476] | Set | GOOSE-Meldung Identifikationsstring. |

6.3.3.3.6 FB_ScsmGoCBServerJobClass

Server-Implementierung des GOCB-Steuerblocks.

Namensraum: [Tc3_iec61850_8_1 \[▶ 97\]](#)

Bibliothek: [Tc3_iec61850_8_1 \(Tc3_iec61850_8_1.compiled-library\)](#)

Vererbungshierarchie

[FB_AsyncServiceResultClass \[▶ 244\]](#) -> [FB_ScsmGoCBServerJobClass](#)

```
FUNCTION_BLOCK FB_ScsmGoCBServerJobClass EXTENDS FB_AsyncServiceResultClass IMPLEMENTS I_ScsmGoCBServerJobClass
```

 Schnittstellen

| Typ | Beschreibung |
|--|--|
| I_ScsmGoCBServerJobClass [▶ 347] | Server-Schnittstelle der GOCB-Steuerblock Implementierung. |

 Methoden

| Name | Beschreibung |
|---------------------------------|--|
| Execute [▶ 347] | Führt die GOCB-Zustandsmaschine des Servers aus. |

6.3.3.3.7 FB_ScsmUrCBImplClass

Der Funktionsbaustein [FB_ScsmUrCBImplClass](#) beinhaltet die Client- und Server-Implementierung der Eigenschaften und Funktionen eines UR CB-Steuerblocks. Für jeden konfigurierten GOOSE-Steuerblock wird vom TwinCAT Telecontrol Configurator eine Instanz von [FB_ScsmUrCBImplClass](#) angelegt.

Namensraum: [Tc3_iec61850_8_1 \[▶ 97\]](#)

Bibliothek: [Tc3_iec61850_8_1 \(Tc3_iec61850_8_1.compiled-library\)](#)

Vererbungshierarchie

FB_AcsiCommonNodeClass [▶ 154] -> FB_AcsiCommonDataClass [▶ 148] -

> FB_AcsiCommonControlBlockClass [▶ 147] -> FB_AcsiCommonUnbufferedReportControlBlockClass [▶ 154] -

> FB_AcsiUnbufferedReportControlBlock [▶ 165] -> FB_ScsmUrCBImplClass

```
FUNCTION_BLOCK FB_ScsmUrCBImplClass EXTENDS FB_AcsiUnbufferedReportControlBlock IMPLEMENTS I_ScsmUrCBImplClass
VAR_INPUT
    Client : FB_ScsmUrCBClientJobClass;
    Server : FB_ScsmUrCBServerJobClass;
END_VAR
```

Schnittstellen

| Typ | Beschreibung |
|-----------------------------|---|
| I_ScsmUrCBImplClass [▶ 348] | Schnittstelle der URCB-Steuerblock Implementierung. |

Eingänge

| Name | Typ | Definitionsort | Beschreibung |
|---------|--|--|---|
| Client | FB_ScsmUrCBClientJobClass [▶ 237] | Local | Client-Implementierung des URCB-Steuerblocks. |
| Server | FB_ScsmUrCBServerJobClass [▶ 238] | Local | Server-Implementierung des URCB-Steuerblocks. |
| RptID | FB_AcsiDATypeVisString129 [▶ 144] | Geerbt von FB_AcsiUnbufferedReportControlBlock [▶ 165] | Datenattribut, Ed1=>RptID:VisStr[65]! |
| RptEna | FB_AcsiBATypeBOOLEAN [▶ 98] | -/- | Datenattribut |
| Resv | FB_AcsiBATypeBOOLEAN [▶ 98] | -/- | Datenattribut |
| DatSet | FB_AcsiDATypeDatSetRef [▶ 146] | -/- | Datenattribut |
| ConfRev | FB_AcsiBATypeINT32U [▶ 101] | -/- | Datenattribut |
| OptFlds | FB_AcsiDATypeOptFlds [▶ 123] | -/- | Datenattribut |
| BufTm | FB_AcsiBATypeINT32U [▶ 101] | -/- | Datenattribut |
| SqNum | FB_AcsiBATypeINT8U [▶ 100] | -/- | Datenattribut |
| TrgOps | FB_AcsiDATypeTriggerConditions [▶ 141] | -/- | Datenattribut |
| IntgPd | FB_AcsiBATypeINT32U [▶ 101] | -/- | Datenattribut |
| GI | FB_AcsiBATypeBOOLEAN [▶ 98] | -/- | Datenattribut |
| Owner | FB_AcsiDATypeOctet64 [▶ 122] | -/- | Datenattribut, Ed2. |

 **Eigenschaften**

| Name | Type | Zu-griff | Definitions-ort | Beschreibung |
|---------------|--|-------------|---|--|
| bGI | BOOL | Get,
Set | I_ScsmUrCBIm-
mplClass
[▶ 348] | Attributwert: Aktivierung des Generalabfrage-Befehls. |
| bResv | BOOL | Get,
Set | --- | Attributwert: Reservierung des RCBs. |
| bRptE-
na | BOOL | Get,
Set | --- | Attributwert: Aktivierung/Deaktivierung des RCB-
Steuerblocks. Weitere Konfiguration benötigt. |
| cOptFl-
ds | ST_AcsiOptionalFi-
elds [▶ 459] | Get,
Set | --- | Attributwert: Konfiguration der optionalen Felder, die mit
einem Report übertragen werden. |
| cTrgO-
ps | ST_AcsiTriggerCo-
nditions [▶ 466] | Get,
Set | --- | Attributwert: Konfiguration der Auslöseoptionen für das
Senden eines Reports. |
| nBufT-
m | DWORD | Get,
Set | --- | Attributwert: Konfiguration der max. Report-Pufferzeit in
Millisekunden. |
| nConf-
Rev | DWORD | Get,
Set | --- | Attributwert: Revisionsnummer der Konfiguration des RCB-
Steuerblocks. |
| nIntgP-
d | DWORD | Get,
Set | --- | Attributwert: Konfiguration der max. Zeit zwischen den
Integrity-Reports in Millisekunden. |
| nSqN-
um | BYTE | Get,
Set | --- | Attributwert: Report-Sequenznummer. |
| oOwn-
er | T_OCTET64
[▶ 479] | Get,
Set | --- | Attributwert: Eigentümer des RCBs. IP-Adresse des Clients
welcher den RCB für sich reserviert hat (Offline-/Online-
Reservierung). |
| sDatS-
et | T_AcsiObjectRefe-
rence [▶ 476] | Get,
Set | --- | Attributwert: Name des verknüpften Data-Sets. |
| sRptl-
D | T_AcsiVisString12-
9 [▶ 476] | Get,
Set | --- | Attributwert: Report-Identifikationsstring. |
| ipDatS-
et | I_AcsiCommonDa-
taSetClass [▶ 313] | Get | --- | Schnittstellenzeiger des verknüpften Data-Sets. |
| ipClie-
nt | I_ScsmUrCBClient-
JobClass [▶ 348] | Get | --- | Schnittstellenzeiger der Client-Implementierung des URCB-
Steuerblocks. |
| ipServ-
er | I_ScsmUrCBServe-
rJobClass [▶ 349] | Get | --- | Schnittstellenzeiger der Server-Implementierung des
URCB-Steuerblocks. |

6.3.3.3.8 FB_ScsmUrCBClientJobClass

Client-Implementierung des URCB-Steuerblocks.

Namensraum: [Tc3_iec61850_8_1](#) [▶ 97]

Bibliothek: [Tc3_iec61850_8_1](#) (Tc3_iec61850_8_1.compiled-library)

Vererbungshierarchie

[FB_AsyncServiceResultClass](#) [▶ 244] -> [FB_ScsmUrCBClientJobClass](#)

```
FUNCTION_BLOCK FB_ScsmUrCBClientJobClass EXTENDS FB_AsyncServiceResultClass IMPLEMENTS I_ScsmUrCBClientJobClass
VAR_OUTPUT
    stInfo : ST_ScsmUrCBJobInfo;
END_VAR
```

Schnittstellen

| Typ | Beschreibung |
|--|--|
| I_ScsmUrCBClientJobClass [▶ 348] | Client-Schnittstelle der URCB-Steuerblock Implementierung. |

Ausgänge

| Name | Typ | Beschreibung |
|--------|--|---|
| stInfo | ST_ScsmUrCBJobInfo [▶ 475] | Information, die in der letzten Report Nachricht empfangen wurde. |

Methoden

| Name | Beschreibung |
|------------------------------------|--------------------------------------|
| EnableReq [▶ 349] | Aktiviert den Steuerblock. |
| DisableReq [▶ 349] | Deaktiviert den Steuerblock. |
| Glreq [▶ 349] | Aktiviert den Generalabfrage-Befehl. |

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|----------|--|---------|--|
| cOptFlds | ST_AcsiOptionalFields [▶ 459] | Set | Optionalen Felder, die in einem Report übertragen werden. |
| cTrgOps | ST_AcsiTriggerConditions [▶ 466] | Set | Auslösoptionen für das Senden eines Reports. |
| nBufTm | DWORD | Set | Max. Report-Pufferzeit in Millisekunden. |
| nIntgPd | DWORD | Set | Max. Zeit zwischen den Integrity-Reports in Millisekunden. |
| sRptID | T_AcsiVisString129 [▶ 476] | Set | Report-Identifikationsstring. |

6.3.3.3.9 FB_ScsmUrCBServerJobClass

Server-Implementierung des URCB-Steuerblocks.

Namensraum: Tc3_iec61850_8_1 [\[▶ 97\]](#)

Bibliothek: Tc3_iec61850_8_1 (Tc3_iec61850_8_1.compiled-library)

Vererbungshierarchie


[FB_AsyncServiceResultClass \[▶ 244\]](#) -> [FB_ScsmUrCBServerJobClass](#)

```
FUNCTION_BLOCK FB_ScsmUrCBServerJobClass EXTENDS FB_AsyncServiceResultClass IMPLEMENTS I_ScsmUrCBServerJobClass
VAR_OUTPUT
    stInfo : ST_ScsmUrCBJobInfo;
END_VAR
```


Schnittstellen

| Typ | Beschreibung |
|--|--|
| I_ScsmUrCBServerJobClass [▶ 349] | Server-Schnittstelle der URCB-Steuerblock Implementierung. |

 **Ausgänge**

| Name | Typ | Beschreibung |
|--------|--|--|
| stInfo | ST_ScsmUrCBJobInfo  475 | Information, die in der letzten Report Nachricht gesendet wurde. |

 **Methoden**

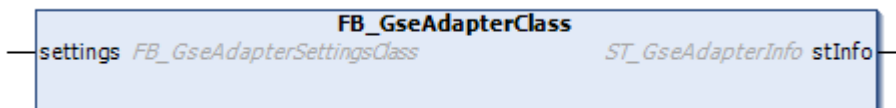
| Name | Beschreibung |
|---|---|
| Execute  350 | Führt die RCB-Zustandsmaschine des Servers aus. |

 **Eigenschaften**


| Name | Typ | Zu-griff | Beschreibung |
|--------------------------------|------------|----------|--|
| bOverwriteSameCycleCh
anges | BOOL | Set | Bei TRUE werden mehrfache Wertänderungen innerhalb eines SPS-Zyklus und eines Datenobjekts als einzelne Wertänderungen behandelt (es wird nur ein Report generiert). Bei FALSE generiert jede Wertänderung einen eigenen Report. |
| bReplaceBufferedMX | BOOL | Set | Konfiguriert, ob Daten der funktionalen Gruppe „MX“, gesondert, wie Daten der funktionalen Gruppe „ST“, behandelt werden sollen. |
| sResvClient | STRIN
G | Set | String mit der IP-Adresse des in SCL vorkonfigurierten Clients (Offline-Reservierung). Beispiel: '192.168.10.140'. Wenn gesetzt, dann wird im Server der Zugriff auf die RCB-Instanz auf den Client mit dieser IP-Adresse beschränkt („ClientLN“-Einträge in der SCL). Default: Leerstring (Online-Reservierung). Beim Leerstring kann jeder Client, dynamisch den Zugriff auf den RCB für sich reservieren. |

6.3.3.4 GOOSE

6.3.3.4.1 FB_GseAdapterClass




Der Funktionsbaustein FB_GseAdapterClass implementiert Schnittstellen zum Austausch der hochprioren GSE-Daten (Global System Event) über eine ausgewählte Netzwerkschnittstelle. Zu diesen Daten gehören GOOSE, GSSE oder SV (Sampled Values) Nachrichten.

Um die Kommunikation ausführen zu können, muss jede Instanz von FB_GseAdapterClass mit einem entsprechenden Netzwerkadapter verbunden werden. Dies ist in [RT - Ethernet Adapter Konfiguration](#)  [371](#) erklärt.

Namensraum: [Tc3_Gse](#)  [97](#)

Bibliothek: Tc3_Gse (Tc3_Gse.compiled-library)

Vererbungshierarchie

[FB_ServiceErrorClass](#)  [245](#) -> FB_GseAdapterClass

```

FUNCTION_BLOCK FB_GseAdapterClass EXTENDS FB_ServiceErrorClass IMPLEMENTS I_GseAdapterClass, I_GseAd
apterEventSink
VAR_INPUT
    settings : FB_GseAdapterSettingsClass;
END_VAR
VAR_OUTPUT
    stInfo : ST_GseAdapterInfo;
END_VAR
  
```

Schnittstellen

| Typ | Beschreibung |
|---|-----------------------------------|
| I_GseAdapterClass [▶ 337] | Schnittstelle zum Datenaustausch. |
| I_GseAdapterEventSink | |

Eingänge

| Name | Typ | Beschreibung |
|----------|--|--------------------------------------|
| settings | FB_GseAdapterSettingsClass [▶ 166] | Adapter-Konfigurationseinstellungen. |

Methoden

| Name | Beschreibung |
|---|---|
| StartPublisher
[▶ 240] | Startet den Publishing-Prozess bei allen GOCBs. |
| StopPublisher
[▶ 241] | Stoppt den Publishing-Prozess bei allen GOCBs. |
| Subscribe
[▶ 241] | Startet den Subscriber-Prozess bei allen GOCBs. |
| Unsubscribe
[▶ 241] | Stoppt den Subscriber-Prozess bei allen GOCBs. |

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|---------------|---|---------|---|
| ipGroup | I_AcsiCommonIEDGroupClass [▶ 315] | Set | Schnittstellenzeiger von einer IED-Gruppe. |
| ipIED | I_AcsiCommonIntelligentElectronicDeviceClass
[▶ 316] | Set | Schnittstellenzeiger von einem Objekt mit der Implementierung der obersten Ebene des IEC 61850 Datenmodells. |
| ipLinkStatus | I_GseLinkStatusEventSink
[▶ 340] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnLinkStatusChange. |
| ipSettings | I_GseAdapterSettingsClass
[▶ 322] | Get | Protokolleinstellungen des GSE-Adapters. Gibt einen Schnittstellenzeiger auf „settings“ zurück. |
| ipSystemClock | I_GseSystemClockEventSink
[▶ 341] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnGetSystemTime. |

Ausgänge

| Name | Typ | Beschreibung |
|--------|---|--|
| stInfo | ST_GseAdapterInfo [▶ 468] | Statistische Information zur GSE-Kommunikation (z.B. über die Anzahl der empfangenen oder versendeten Telegramme). |

6.3.3.4.1 StartPublisher

Startet den Publishing-Prozess bei allen GOCBs.

```
METHOD FINAL StartPublisher : BOOL
VAR_OUTPUT
    ipError : I_ServiceErrorClass;
END_VAR
```

ipError: Schnittstellenzeiger vom Typ [I_ServiceErrorClass](#) [► 392]. Über diesen Zeiger kann bei einer negativen Rückmeldung der Fehlercode, Fehlerquelle oder Fehlertext abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) beim Erfolg oder negative Rückmeldung (FALSE) beim Fehler.

6.3.3.4.1.2 StopPublisher

Stoppt den Publishing-Prozess bei allen GOCBs.

```
METHOD FINAL StopPublisher : BOOL
VAR_OUTPUT
    ipError : I_ServiceErrorClass;
END_VAR
```

ipError: Schnittstellenzeiger vom Typ [I_ServiceErrorClass](#) [► 392]. Über diesen Zeiger kann bei einer negativen Rückmeldung der Fehlercode, Fehlerquelle oder Fehlertext abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) beim Erfolg oder negative Rückmeldung (FALSE) beim Fehler.

6.3.3.4.1.3 Subscribe

Startet den Subscriber-Prozess bei allen GOCBs.

```
METHOD FINAL Subscribe : BOOL
VAR_OUTPUT
    ipError : I_ServiceErrorClass;
END_VAR
```

ipError: Schnittstellenzeiger vom Typ [I_ServiceErrorClass](#) [► 392]. Über diesen Zeiger kann bei einer negativen Rückmeldung der Fehlercode, Fehlerquelle oder Fehlertext abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) beim Erfolg oder negative Rückmeldung (FALSE) beim Fehler.

6.3.3.4.1.4 Unsubscribe

Stoppt den Subscriber-Prozess bei allen GOCBs.

```
METHOD FINAL Unsubscribe : BOOL
VAR_OUTPUT
    ipError : I_ServiceErrorClass;
END_VAR
```

ipError: Schnittstellenzeiger vom Typ [I_ServiceErrorClass](#) [► 392]. Über diesen Zeiger kann bei einer negativen Rückmeldung der Fehlercode, Fehlerquelle oder Fehlertext abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) beim Erfolg oder negative Rückmeldung (FALSE) beim Fehler.

6.3.3.4.2 FB_GseGoCBImplClass

Der Funktionsbaustein FB_GseGoCBImplClass beinhaltet die Implementierung der Eigenschaften eines GOOSE-Steuerblocks sowie die Funktionen für Publisher und Subscriber. Für jeden konfigurierten GOOSE-Steuerblock wird vom TwinCAT Telecontrol Configurator eine Instanz von FB_GseGoCBImplClass angelegt. Für weitere Informationen zur Code-Erzeugung für GOOSE siehe [Automatische Code-Erzeugung](#) [► 74].

Namensraum: [Tc3_Gse](#) [► 97]

Bibliothek: Tc3_Gse (Tc3_Gse.compiled-library)

Vererbungshierarchie

FB_AcsiCommonNodeClass [▶ 154] -> FB_AcsiCommonDataClass [▶ 148] -
 > FB_AcsiCommonControlBlockClass [▶ 147] -> FB_AcsiCommonGooseControlBlockClass [▶ 150] -
 > FB_AcsiGooseControlBlock [▶ 165] -> FB_ScsmGoCBImplClass [▶ 232] -> FB_GseGoCBImplClass

```
FUNCTION_BLOCK FB_GseGoCBImplClass EXTENDS FB_ScsmGoCBImplClass IMPLEMENTS I_GseGoCBImplClass
VAR_INPUT
    Publisher : FB_GseGoCBPublisherClass;
    Subscriber : FB_GseGoCBSubscriberClass;
END_VAR
```

Schnittstellen

| Typ | Beschreibung |
|----------------------------|--|
| I_GseGoCBImplClass [▶ 338] | Schnittstelle der GOOSE-Steuerblock Implementierung. |

Eingänge

| Name | Typ | Definitionsort | Beschreibung |
|------------|------------------------------------|---|--|
| Publisher | FB_GseGoCBPublisherClass [▶ 243] | Local | Publisher-Implementierung des GOOSE-Steuerblocks. |
| Subscriber | FB_GseGoCBSubscriberClass [▶ 243] | Local | Subscriber-Implementierung des GOOSE-Steuerblocks. |
| Client | FB_ScsmGoCBClientJobClass [▶ 234] | Geerbt von FB_ScsmGoCBImplClass [▶ 232] | Client-Implementierung des GOOSE-Steuerblocks. |
| Server | FB_ScsmGoCBServerJobClass [▶ 235] | -/- | Server-Implementierung des GOOSE-Steuerblocks. |
| GoEna | FB_AcsiBATypeBOOLEAN [▶ 98] | Geerbt von FB_AcsiGooseControlBlock [▶ 165] | Datenattribut |
| GoID | FB_AcsiDATypeVisString129 [▶ 144] | -/- | Datenattribut |
| DatSet | FB_AcsiDATypeDatSetRef [▶ 146] | -/- | Datenattribut |
| ConfRev | FB_AcsiBATypeINT32U [▶ 101] | -/- | Datenattribut |
| NdsCom | FB_AcsiBATypeBOOLEAN [▶ 98] | -/- | Datenattribut |
| DstAddress | FB_AcsiDATypePhyComAddress [▶ 127] | -/- | Datenattribut |
| MinTime | FB_AcsiBATypeINT32U [▶ 101] | -/- | Datenattribut |
| MaxTime | FB_AcsiBATypeINT32U [▶ 101] | -/- | Datenattribut |
| FixedOffs | FB_AcsiBATypeBOOLEAN [▶ 98] | -/- | Datenattribut |

Beispiel

Für weitere Informationen über die Anwendung des GOOSE-Steuerblocks siehe: Client - GOOSE Subscriber (ohne Client-Server-Kommunikation) [▶ 622] und Server - GOOSE Publisher (ohne Client-Server-Kommunikation) [▶ 633].

6.3.3.4.3 FB_GseGoCBPublisherClass

Publisher-Implementierung des GOOSE-Steuerblocks.

Namensraum: [Tc3_Gse \[▶ 97\]](#)

Bibliothek: Tc3_Gse (Tc3_Gse.compiled-library)

Vererbungshierarchie

[FB_ServiceErrorClass \[▶ 245\]](#) -> [FB_GseGoCBPublisherClass](#)

```
FUNCTION_BLOCK FB_GseGoCBPublisherClass EXTENDS FB_ServiceErrorClass IMPLEMENTS I_GseGoCBPublisherClass
VAR_OUTPUT
    stTx : ST_GseGoCBPublisherTx;
END_VAR
```

Schnittstellen

| Typ | Beschreibung |
|---|--|
| I_GseGoCBPublisherClass [▶ 338] | Schnittstelle der GOOSE-Steuerblock Implementierung. |

Ausgänge

| Name | Typ | Beschreibung |
|------|---|---|
| stTx | ST_GseGoCBPublisherTx [▶ 468] | Information, die in der letzten GOOSE-Meldung gesendet wurde. |

Methoden

| Name | Beschreibung |
|---------------------------------|--|
| Execute [▶ 338] | Führt die Publisher-Zustandsmaschine aus. |
| Start [▶ 339] | Startet das Versenden von GOOSE-Nachrichten. |
| Stop [▶ 339] | Stoppt das Versenden von GOOSE-Nachrichten. |
| Update [▶ 339] | Löst ein sofortiges Versenden einer GOOSE-Nachricht aus. |

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|-------------|---|---------|--|
| bTest | BOOL | Set | Simulation/Test-Flag der GOOSE-Meldungen. Beim Wert TRUE wird der Simulation- und Test-Flag in der GOOSE-Meldung gesetzt. |
| eStrategy | E_GseRetransmissionStrategy [▶ 447] | Set | Konfiguriert den Algorithmus für die wiederholten GOOSE-Meldungen. |
| nMultiplier | BYTE(1..16#FF) | Set | Konfiguriert den Multiplikator für den Zeitintervall der wiederholten GOOSE-Meldungen. Defaultwert: Param_Acsi.cDefault_GoCB_Multiplier (2). |
| ipAdapter | I_GseAdapterClass [▶ 337] | Set | Schnittstelle des GOOSE-Netzwerkadapters. |

6.3.3.4.4 FB_GseGoCBSubscriberClass

Subscriber-Implementierung des GOOSE-Steuerblocks.

Namensraum: [Tc3_Gse \[▶ 97\]](#)

Bibliothek: Tc3_Gse (Tc3_Gse.compiled-library)

Vererbungshierarchie

FB_ServiceErrorClass [▶ 245] -> FB_GseGoCBSsubscriberClass

```
FUNCTION_BLOCK FB_GseGoCBSsubscriberClass EXTENDS FB_ServiceErrorClass IMPLEMENTS I_GseGoCBSsubscriberClass
VAR_OUTPUT
    stRx : ST_GseGoCBSsubscriberRx;
END_VAR
```

Schnittstellen

| Typ | Beschreibung |
|-----------------------------------|--|
| I_GseGoCBSsubscriberClass [▶ 339] | Schnittstelle der GOOSE-Steuerblock Implementierung. |

Ausgänge

| Name | Typ | Beschreibung |
|------|---------------------------------|--|
| stTx | ST_GseGoCBSsubscriberRx [▶ 470] | Information, die in der letzten GOOSE-Meldung empfangen wurde. |

Methoden

| Name | Beschreibung |
|-----------------|--|
| Execute [▶ 340] | Führt die Subscriber-Zustandsmaschine aus. |
| Enable [▶ 340] | Startet das Empfangen von GOOSE-Nachrichten. |
| Disable [▶ 340] | Stoppt das Empfangen von GOOSE-Nachrichten. |

6.3.4 FB_AsyncServiceResultClass

Der Funktionsbaustein FB_AsyncServiceResultClass implementiert die I_AsyncServiceResultClass [▶ 387]-Schnittstelle zur Statusabfrage und Verfolgung von asynchronen Vorgängen. Die Methoden des Kommunikationsbausteins FB_iec61850CommonClass [▶ 220] zum Aufbau/Abbau der Verbindung und zur Aktivierung der IEC 61850 Dienste liefern den Schnittstellenzeiger vom Typ I_AsyncServiceResultClass [▶ 387] als Parameter zurück. Über diesen Zeiger kann die Applikation den Status/Fortgang der Dienstauführung überwachen und abfragen.

Namensraum: Tc3_Collections [▶ 96]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Vererbungshierarchie

FB_AsyncServiceResultClass

```
FUNCTION_BLOCK FB_AsyncServiceResultClass IMPLEMENTS I_AsyncServiceResultClass
```

Schnittstellen

| Typ | Beschreibung |
|-----------------------------------|---|
| I_AsyncServiceResultClass [▶ 387] | Schnittstelle zur Statusabfrage und Verfolgung von asynchronen Vorgängen. |

 Methoden

| Name | Beschreibung |
|--------------------------------------|---|
| CopyFrom [▶ 388] | Kopiert Objektdaten von einer anderen Objektinstanz. |
| Dump [▶ 388] | Konvertiert eigene Objektdaten in einen formatierten String. |
| Equal [▶ 388] | Wertevergleich der eigenen Objektdaten mit externen Datenwerten. |
| EqualEx [▶ 389] | Wertevergleich der eigenen Objektdaten mit externen Datenwerten. |
| EqualTo [▶ 389] | Wertevergleich der eigenen Objektdaten mit den Daten einer anderen Objektinstanz. |
| IsAborted [▶ 389] | Prüft, ob die Dienstaufführung abgebrochen wurde. |
| IsBusy [▶ 389] | Prüft, ob der Dienst mit der Ausführung noch beschäftigt ist. |
| IsCompleted [▶ 390] | Prüft, ob die Dienstaufführung erfolgreich abgeschlossen wurde. |
| IsFailed [▶ 390] | Prüft, ob die Dienstaufführung mit einem Fehler abgeschlossen wurde. |
| SetAborted [▶ 390] | Setzt den Abbruchstatus. |
| SetAbortedEx [▶ 390] | Setzt den Abbruchstatus. |
| SetBusy [▶ 391] | Setzt den Status auf Beschäftigt. |
| SetCompleted [▶ 391] | Setzt den Status auf Erledigt (ohne Fehler). |
| SetFailed [▶ 391] | Setzt den Fehlerstatus. |
| SetFailedEx [▶ 392] | Setzt den Fehlerstatus. |

 Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------------------------------|---|---------|--------------------|
| eState [▶ 392] | E AsyncServiceState [▶ 444] | Get | Statusinformation. |

6.3.5 FB_ServiceErrorClass

Der Funktionsbaustein FB_ServiceErrorClass implementiert die [I_ServiceErrorClass \[▶ 392\]](#)-Schnittstelle zur Fehlerabfrage und Fehlerspeicherung. Die Methoden der Objektinstanzen des TwinCAT IEC 61850 Datenmodells und einige wenige Dienste zum Aufbau/Abbau der Verbindung liefern beim Fehler den Schnittstellenzeiger vom Typ [I_ServiceErrorClass \[▶ 392\]](#). Über diesen Zeiger kann eine genauere Fehleranalyse durchgeführt werden.

Namensraum: [Tc3_Collections \[▶ 96\]](#)

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Vererbungshierarchie

FB_ServiceErrorClass

```
FUNCTION_BLOCK FB_ServiceErrorClass IMPLEMENTS I_ServiceErrorClass
```

 Schnittstellen

| Typ | Beschreibung |
|---|--|
| I_ServiceErrorClass [▶ 392] | Schnittstelle zur Fehlerabfrage und Fehlerspeicherung. |

Methoden

| Name | Beschreibung |
|--------------------------------------|---|
| CopyFrom [▶ 393] | Kopiert Objektdaten von einer anderen Objektinstanz. |
| Dump [▶ 393] | Konvertiert eigene Objektdaten in einen formatierten String (z.B. für Logausgaben). |
| Equal [▶ 393] | Wertevergleich der eigenen Objektdaten (Fehlerinformation) mit externen Datenwerten. |
| EqualEx [▶ 394] | Wertevergleich der eigenen Objektdaten (Fehlerinformation) mit externen Datenwerten. |
| EqualTo [▶ 394] | Wertevergleich der eigenen Objektdaten (Fehlerinformation) mit den Daten einer anderen Objektinstanz. |
| IsFailed [▶ 394] | Prüft eigene Objektdaten (Fehlerinformation) auf Misserfolg. |
| IsSucceeded [▶ 394] | Prüft eigene Objektdaten (Fehlerinformation) auf Erfolg. |
| SetFailedEx [▶ 394] | Speichert Fehlerinformation in der eigenen Objektinstanz. |
| SetSucceeded [▶ 395] | Speichert Erfolgsinformation in der eigenen Objektinstanz. |

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|---------------------------------|--|---------|-------------------------------------|
| stError [▶ 395] | ST_ServiceErrorClass [▶ 475] | Get | Objektdaten als strukturierter Typ. |

6.4 Funktionen

6.4.1 Accuracy_To_UtcTimeQualityAccuracy

Konvertiert den Wert des Aufzählungstyps [E_UtcTimeAccuracy \[▶ 449\]](#) (Genauigkeit der Zeitinformation) in fünf boolische Werte. Diese fünf Ausgangswerte entsprechen den Bit-Komponenten: Accuracy0..Accuracy4 des strukturierten Typs [T_UtcTimeQuality \[▶ 480\]](#).

Namensraum: [Tc3_Collections \[▶ 96\]](#)

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
FUNCTION Accuracy_To_UtcTimeQualityAccuracy
VAR_INPUT
    in          : E_UtcTimeAccuracy;
END_VAR
VAR_OUTPUT
    bAccuracy0 : BOOL;
    bAccuracy1 : BOOL;
    bAccuracy2 : BOOL;
    bAccuracy3 : BOOL;
    bAccuracy4 : BOOL;
END_VAR
```

Eingänge

| Name | Typ | Beschreibung |
|------|---|---|
| in | E_UtcTimeAccuracy [▶ 449] | Genauigkeit des Bruchteils der Sekunde. Dieser Wert bestimmt die max. Anzahl der signifikanten Bits der T_UtcTime [▶ 479] .fractionOfSecond-Komponente. |

 **Ausgänge**

Konvertierte Information als [T_UtcTimeQuality](#) [[▶ 480](#)].Accuracy0..4 Bit-Komponenten. Siehe: [Kodierung der UTC-Time.Quality.Accuracy0..4-Bits](#) [[▶ 683](#)].

| Name | Typ | Beschreibung |
|------------|------|------------------------|
| bAccuracy0 | BOOL | Genauigkeit MSB-Bit 0. |
| bAccuracy1 | BOOL | Genauigkeit Bit 1. |
| bAccuracy2 | BOOL | Genauigkeit Bit 2. |
| bAccuracy3 | BOOL | Genauigkeit Bit 3. |
| bAccuracy4 | BOOL | Genauigkeit LSB-Bit 4. |

 **Rückgabewert**

Keiner.

Beispiel

```

METHOD UtcTimeQualityAccuracy_ToFrom_Accuracy : BOOL
VAR_INPUT
    fbIED      : REFERENCE TO FB_IED;
END_VAR
VAR
    Quality   : T_UtcTimeQuality;
    sQuality  : STRING;
END_VAR

IF NOT __ISVALIDREF(fbIED) THEN
    RETURN;
END_IF

Quality.LeapSecondsKnown := FALSE;
Quality.ClockFailure     := FALSE;
Quality.ClockNotSynchronized := FALSE;
Accuracy_To_UtcTimeQualityAccuracy( E_UtcTimeAccuracy.Null, (* Null bits accuracy *)
    bAccuracy0=>Quality.Accuracy0,
    bAccuracy1=>Quality.Accuracy1,
    bAccuracy2=>Quality.Accuracy2,
    bAccuracy3=>Quality.Accuracy3,
    bAccuracy4=>Quality.Accuracy4 );
sQuality := UtcTimeQuality_To_String(Quality); (* sQuality := 'UQ#000|0' *)
IF fbIED.Relay.LLN0.Beh.t.eAccuracy <> UtcTimeQualityAccuracy_To_Accuracy(Quality) THEN
    Quality := fbIED.Relay.LLN0.Beh.t.Quality;
END_IF

Quality.LeapSecondsKnown := FALSE;
Quality.ClockFailure     := FALSE;
Quality.ClockNotSynchronized := FALSE;
Accuracy_To_UtcTimeQualityAccuracy( E_UtcTimeAccuracy._03, (* 3 bits accuracy *)
    bAccuracy0=>Quality.Accuracy0,
    bAccuracy1=>Quality.Accuracy1,
    bAccuracy2=>Quality.Accuracy2,
    bAccuracy3=>Quality.Accuracy3,
    bAccuracy4=>Quality.Accuracy4 );
sQuality := UtcTimeQuality_To_String(Quality); (* sQuality := 'UQ#000|3' *)
IF fbIED.Relay.LLN0.Health.t.eAccuracy <> UtcTimeQualityAccuracy_To_Accuracy(Quality) THEN
    Quality := fbIED.Relay.LLN0.Health.t.Quality;
END_IF

Quality.LeapSecondsKnown := FALSE;
Quality.ClockFailure     := FALSE;
Quality.ClockNotSynchronized := FALSE;
Accuracy_To_UtcTimeQualityAccuracy( E_UtcTimeAccuracy._05, (* 5 bits accuracy *)
    bAccuracy0=>Quality.Accuracy0,
    bAccuracy1=>Quality.Accuracy1,
    bAccuracy2=>Quality.Accuracy2,
    bAccuracy3=>Quality.Accuracy3,
    bAccuracy4=>Quality.Accuracy4 );
sQuality := UtcTimeQuality_To_String(Quality); (* sQuality := 'UQ#000|5' *)
IF fbIED.Relay.LLN0.Mod_.t.eAccuracy <> UtcTimeQualityAccuracy_To_Accuracy(Quality) THEN
    Quality := fbIED.Relay.LLN0.Mod_.t.Quality;
END_IF
    
```

```

Quality.LeapSecondsKnown := FALSE;
Quality.ClockFailure := FALSE;
Quality.ClockNotSynchronized := FALSE;
Accuracy_To_UtcTimeQualityAccuracy( E_UtcTimeAccuracy.Unspecified, (* Unspecified accuracy *)
    bAccuracy0=>Quality.Accuracy0,
    bAccuracy1=>Quality.Accuracy1,
    bAccuracy2=>Quality.Accuracy2,
    bAccuracy3=>Quality.Accuracy3,
    bAccuracy4=>Quality.Accuracy4 );
sQuality := UtcTimeQuality_To_String(Quality); (* sQuality := 'UQ#000|31' *)
IF fbIED.Relay.MMXU1.phV.phsA.t.eAccuracy <> UtcTimeQualityAccuracy_To_Accuracy(Quality) THEN
    Quality := fbIED.Relay.MMXU1.phV.phsA.t.Quality;
END_IF

UtcTimeQualityAccuracy_ToFrom_Accuracy := TRUE;

```

6.4.2 AcsiAnalogueValue

Initialisiert die Datenelemente der Struktur: ST_AcsiAnalogueValue.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

```

FUNCTION AcsiAnalogueValue : ST_AcsiAnalogueValue
VAR_INPUT
    i : DINT;
    f : REAL;
END_VAR

```

Eingänge

| Name | Typ | Beschreibung |
|------|------|--|
| i | DINT | Datenelement „i“, das gesetzt werden soll. |
| f | REAL | Datenelement „f“, das gesetzt werden soll. |

Rückgabewert

| Name | Typ | Beschreibung |
|-------------------|--|--|
| AcsiAnalogueValue | ST_AcsiAnalogueValue [▶ 455] | Struktur mit initialisierten Datenelementen. |

Beispiel

```

METHOD FINAL Sample_AcsiAnalogueValue : BOOL
VAR
    mag: ST_AcsiAnalogueValue;
END_VAR

mag:=AcsiAnalogueValue(i:=0, f:=12.45);
Sample_AcsiAnalogueValue:=TRUE;

```

6.4.3 AcsiCalendarTime

Initialisiert die Datenelemente der Struktur: ST_AcsiCalendarTime.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

```

FUNCTION AcsiCalendarTime : ST_AcsiCalendarTime
VAR_INPUT
    occ      : WORD;
    occType  : E_AcsiOccType;
    occPer   : E_AcsiOccPer;
    weekDay  : E_AcsiWeekDay;
    month    : E_AcsiMonth;
    day      : BYTE;
    hr       : BYTE;
    mn       : BYTE;
END_VAR

```

 **Eingänge**

| Name | Typ | Beschreibung |
|---------|---------------------------------------|--|
| occ | WORD | Datenelement „occ“, das gesetzt werden soll. |
| occType | E_AcsiOccType [▶ 417] | Datenelement „occType“, das gesetzt werden soll. |
| occPer | E_AcsiOccPer [▶ 417] | Datenelement „occPer“, das gesetzt werden soll. |
| weekDay | E_AcsiWeekDay [▶ 442] | Datenelement „weekDay“, das gesetzt werden soll. |
| month | E_AcsiMonth [▶ 415] | Datenelement „month“, das gesetzt werden soll. |
| day | BYTE | Datenelement „day“, das gesetzt werden soll. |
| hr | BYTE | Datenelement „hr“, das gesetzt werden soll. |
| mn | BYTE | Datenelement „mn“, das gesetzt werden soll. |

 **Rückgabewert**

| Name | Typ | Beschreibung |
|------------------|---|--|
| AcsiCalendarTime | ST_AcsiCalendarTime [▶ 455] | Struktur mit initialisierten Datenelementen. |

Beispiel

```

METHOD FINAL Sample_AcsiCalendarTime : BOOL
VAR
    calTm: ST_AcsiCalendarTime;
END_VAR

calTm:=AcsiCalendarTime(occ:=0, occType:=E_AcsiOccType.Time_,
    occPer:=E_AcsiOccPer.Hour, weekDay:=E_AcsiWeekDay.Friday,
    month:=E_AcsiMonth.April, day:=1, hr:=0, mn:=0);
Sample_AcsiCalendarTime:=TRUE;
    
```

6.4.4 AcsiCell

Initialisiert die Datenelemente der Struktur: ST_AcsiCell.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

```

FUNCTION AcsiCell : ST_AcsiCell
VAR_INPUT
    xStart : REAL;
    xEnd : REAL;
    yStart : REAL;
    yEnd : REAL;
END_VAR
    
```

 **Eingänge**

| Name | Typ | Beschreibung |
|--------|------|---|
| xStart | REAL | Datenelement „xStart“, das gesetzt werden soll. |
| xEnd | REAL | Datenelement „xEnd“, das gesetzt werden soll. |
| yStart | REAL | Datenelement „yStart“, das gesetzt werden soll. |
| yEnd | REAL | Datenelement „yEnd“, das gesetzt werden soll. |

 **Rückgabewert**

| Name | Typ | Beschreibung |
|----------|-------------------------------------|--|
| AcsiCell | ST_AcsiCell [▶ 455] | Struktur mit initialisierten Datenelementen. |

Beispiel

```
METHOD FINAL Sample_AcsiCell : BOOL
VAR
    cell: ST_AcsiCell;
END_VAR

cell:=AcsiCell(xStart:=1.0, xEnd:=1.0, yStart:=1.0, yEnd:=1.0);
Sample_AcsiCell:=TRUE;
```

6.4.5 AcsiLogOptionalFields

Initialisiert die Datenelemente der Struktur: ST_AcsiLogOptionalFields.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

```
FUNCTION AcsiLogOptionalFields : ST_AcsiLogOptionalFields
VAR_INPUT
    ReasonForInclusion : BOOL;
END_VAR
```

📌 Eingänge

| Name | Typ | Beschreibung |
|--------------------|------|---|
| ReasonForInclusion | BOOL | Datenelement „ReasonForInclusion“, das gesetzt werden soll. |

📌 Rückgabewert

| Name | Typ | Beschreibung |
|-----------------------|---|--|
| AcsiLogOptionalFields | ST_AcsiLogOptionalFields
▶ 459 | Struktur mit initialisierten Datenelementen. |

Beispiel

```
METHOD FINAL Sample_AcsiLogOptionalFields : BOOL
VAR
    fields : ST_AcsiLogOptionalFields;
END_VAR

fields:=AcsiLogOptionalFields(ReasonForInclusion:=FALSE);
Sample_AcsiLogOptionalFields:=TRUE;
```

6.4.6 AcsiOptionalFields

Initialisiert die Datenelemente der Struktur: ST_AcsiOptionalFields.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

```
FUNCTION AcsiOptionalFields : ST_AcsiOptionalFields
VAR_INPUT
    SequenceNumber      : BOOL;
    ReportTimeStamp     : BOOL;
    ReasonForInclusion    : BOOL;
    DataSetName         : BOOL;
    DataReference        : BOOL;
    BufferOverflow       : BOOL;
    EntryID             : BOOL;
    ConfRevision        : BOOL;
    Segmentation        : BOOL;
END_VAR
```

 **Eingänge**

| Name | Typ | Beschreibung |
|--------------------|------|---|
| SequenceNumber | BOOL | Datenelement „SequenceNumber“, das gesetzt werden soll. |
| ReportTimeStamp | BOOL | Datenelement „ReportTimeStamp“, das gesetzt werden soll. |
| ReasonForInclusion | BOOL | Datenelement „ReasonForInclusion“, das gesetzt werden soll. |
| DataSetName | BOOL | Datenelement „DataSetName“, das gesetzt werden soll. |
| DataReference | BOOL | Datenelement „DataReference“, das gesetzt werden soll. |
| BufferOverflow | BOOL | Datenelement „BufferOverflow“, das gesetzt werden soll. |
| EntryID | BOOL | Datenelement „EntryID“, das gesetzt werden soll. |
| ConfRevision | BOOL | Datenelement „ConfRevision“, das gesetzt werden soll. |
| Segmentation | BOOL | Datenelement „Segmentation“, das gesetzt werden soll. |

 **Rückgabewert**

| Name | Typ | Beschreibung |
|--------------------|--|--|
| AcsiOptionalFields | ST_AcsiOptionalFields
▶ 459 | Struktur mit initialisierten Datenelementen. |

Beispiel

```
METHOD FINAL Sample_AcsiOptionalFields : BOOL
VAR
    fields : ST_AcsiOptionalFields;
END_VAR

fields:=AcsiOptionalFields(SequenceNumber:=FALSE, ReportTimeStamp:=FALSE,
    ReasonForInclusion:=FALSE, DataSetName:=FALSE, DataReference:=FALSE,
    BufferOverflow:=FALSE, EntryID:=FALSE, ConfRevision:=0, Segmentation:=FALSE);
Sample_AcsiOptionalFields:=TRUE;
```

6.4.7 AcsiOriginator

Initialisiert die Datenelemente der Struktur: ST_AcsiOriginator.

Namensraum: [Tc3_Acsi ▶ 96](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

```
FUNCTION AcsiOriginator : ST_AcsiOriginator
VAR_INPUT
    orCat : E_AcsiOrCategory;
    orIdent : T_OCTET64;
END_VAR
```

 **Eingänge**

| Name | Typ | Beschreibung |
|---------|---|--|
| orCat | E_AcsiOrCategory
▶ 419 | Datenelement „orCat“, das gesetzt werden soll. |
| orIdent | T_OCTET64 ▶ 479 | Datenelement „orIdent“, das gesetzt werden soll. |

 **Rückgabewert**

| Name | Typ | Beschreibung |
|----------------|---|--|
| AcsiOriginator | ST_AcsiOriginator ▶ 460 | Struktur mit initialisierten Datenelementen. |

Beispiel

```

METHOD FINAL Sample_AcsiOriginator : BOOL
VAR
  org: ST_AcsiOriginator;
  orIdent : T_OCTET64:=[16#36,16#31, 16#38, 16#35, 16#30];
END_VAR

org:=AcsiOriginator(orCat:=E_AcsiOrCategory.StationControl, orIdent:=orIdent);
Sample_AcsiOriginator:=TRUE;

```

6.4.8 AcsiPhyComAddr

Initialisiert die Datenelemente der Struktur: ST_AcsiPhyComAddr.

Namensraum: [Tc3_Acsi \[► 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

```

FUNCTION AcsiPhyComAddr : ST_AcsiPhyComAddr
VAR_INPUT
  Addr      : T_OCTET6;
  PRIORITY  : BYTE;
  VID       : WORD;
  APPID     : WORD;
END_VAR

```

📁 Eingänge

| Name | Typ | Beschreibung |
|----------|----------------------------------|---|
| Addr | T_OCTET6 [► 478] | Datenelement „Addr“, das gesetzt werden soll. |
| PRIORITY | BYTE | Datenelement „PRIORITY“, das gesetzt werden soll. |
| VID | WORD | Datenelement „VID“, das gesetzt werden soll. |
| APPID | WORD | Datenelement „APPID“, das gesetzt werden soll. |

📁 Rückgabewert

| Name | Typ | Beschreibung |
|----------------|---|--|
| AcsiPhyComAddr | ST_AcsiPhyComAddr [► 460] | Struktur mit initialisierten Datenelementen. |

Beispiel

```

METHOD FINAL Sample_AcsiPhyComAddr : BOOL
VAR
  addr: ST_AcsiPhyComAddr;
END_VAR

addr:=AcsiPhyComAddr(Addr:=LWORD_TO_OCTET6(16#AC3040506001), PRIORITY:=4, VID:=16#0001, APPID:=16#0001);
Sample_AcsiPhyComAddr:=TRUE;

```

6.4.9 AcsiPoint

Initialisiert die Datenelemente der Struktur: ST_AcsiPoint.

Namensraum: [Tc3_Acsi \[► 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

```

FUNCTION AcsiPoint : ST_AcsiPoint
VAR_INPUT
  xVal : REAL;
  yVal : REAL;
  zVal : REAL;
END_VAR

```


 **Eingänge**

| Name | Typ | Beschreibung |
|------|------|---|
| xVal | REAL | Datenelement „xVal“, das gesetzt werden soll. |
| yVal | REAL | Datenelement „yVal“, das gesetzt werden soll. |
| zVal | REAL | Datenelement „zVal“, das gesetzt werden soll. |

 **Rückgabewert**

| Name | Typ | Beschreibung |
|------------|---|--|
| AcsciPoint | ST_AcsciPoint [▶ 461] | Struktur mit initialisierten Datenelementen. |

Beispiel

```
METHOD FINAL Sample_AcsciPoint : BOOL
VAR
    point: ST_AcsciPoint;
END_VAR

point:=AcsciPoint(xVal:=1.0, yVal:=1.0, zVal:=1.0);
Sample_AcsciPoint:=TRUE;
```

6.4.10 AcsciPulseConfig

Initialisiert die Datenelemente der Struktur: ST_AcsciPulseConfig.

Namensraum: [Tc3_Acsci](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsci (Tc3_Acsci.compiled-library)

```
FUNCTION AcsciPulseConfig : ST_AcsciPulseConfig
VAR_INPUT
    cmdQual : E_AcsciCmdQual;
    onDur : DWORD;
    offDur : DWORD;
    numPls : DWORD;
END_VAR
```

 **Eingänge**

| Name | Typ | Beschreibung |
|---------|---|--|
| cmdQual | E_AcsciCmdQual
[▶ 404] | Datenelement „cmdQual“, das gesetzt werden soll. |
| onDur | DWORD | Datenelement „onDur“, das gesetzt werden soll. |
| offDur | DWORD | Datenelement „offDur“, das gesetzt werden soll. |
| numPls | DWORD | Datenelement „numPls“, das gesetzt werden soll. |

 **Rückgabewert**

| Name | Typ | Beschreibung |
|------------------|---|--|
| AcsciPulseConfig | ST_AcsciPulseConfig [▶ 462] | Struktur mit initialisierten Datenelementen. |

Beispiel

```
METHOD FINAL Sample_AcsciPulseConfig : BOOL
VAR
    config: ST_AcsciPulseConfig;
END_VAR

config:=AcsciPulseConfig(cmdQual:=E_AcsciCmdQual.Pulse,
    onDur:=10, offDur:=10, numPls:=1);
Sample_AcsciPulseConfig:=TRUE;
```

6.4.11 AcsiQuality

Initialisiert die Datenelemente der Struktur: ST_AcsiQuality.

Namensraum: Tc3_Acsi [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

```
FUNCTION AcsiQuality : ST_AcsiQuality
VAR_INPUT
    Validity0      : BOOL;
    Validity1      : BOOL;
    Overflow       : BOOL;
    OutOfRange     : BOOL;
    BadReference   : BOOL;
    Oscillatory    : BOOL;
    Failure        : BOOL;
    OldData        : BOOL;
    Inconsistent   : BOOL;
    Inaccurate     : BOOL;
    Source         : BOOL;
    Test           : BOOL;
    OperatorBlocked : BOOL;
END_VAR
```

Eingänge

| Name | Typ | Beschreibung |
|-----------------|------|--|
| Validity0 | BOOL | Datenelement „Validity0“, das gesetzt werden soll. |
| Validity1 | BOOL | Datenelement „Validity1“, das gesetzt werden soll. |
| Overflow | BOOL | Datenelement „Overflow“, das gesetzt werden soll. |
| OutOfRange | BOOL | Datenelement „OutOfRange“, das gesetzt werden soll. |
| BadReference | BOOL | Datenelement „BadReference“, das gesetzt werden soll. |
| Oscillatory | BOOL | Datenelement „Oscillatory“, das gesetzt werden soll. |
| Failure | BOOL | Datenelement „Failure“, das gesetzt werden soll. |
| OldData | BOOL | Datenelement „OldData“, das gesetzt werden soll. |
| Inconsistent | BOOL | Datenelement „Inconsistent“, das gesetzt werden soll. |
| Inaccurate | BOOL | Datenelement „Inaccurate“, das gesetzt werden soll. |
| Source | BOOL | Datenelement „Source“, das gesetzt werden soll. |
| Test | BOOL | Datenelement „Test“, das gesetzt werden soll. |
| OperatorBlocked | BOOL | Datenelement „OperatorBlocked“, das gesetzt werden soll. |

Rückgabewert

| Name | Typ | Beschreibung |
|-------------|--|--|
| AcsiQuality | ST_AcsiQuality [▶ 463] | Struktur mit initialisierten Datenelementen. |

Beispiel

```
METHOD FINAL Sample_AcsiQuality : BOOL
VAR
    q: ST_AcsiQuality;
END_VAR

q:=AcsiQuality(Validity0:=0, Validity1:=0, Overflow:=0, OutOfRange:=0,
    BadReference:=0, Oscillatory:=0, Failure:=0, OldData:=0,
    Inconsistent:=0, Inaccurate:=0, Source:=0, Test:=0,
    OperatorBlocked:=0);
Sample_AcsiQuality:=TRUE;
```

6.4.12 AcsiQuality_To_String

Konvertiert die Komponenten des strukturierten Typs [ST_AcsiQuality](#) [▶ 463] in einen formatierten Quality-String. Die Zeichenfolge des resultierenden Strings hat folgenden Aufbau: 'Q#vv|ORBSFDCA|s|TB'. Sie entspricht der TwinCAT [Quality-String-Formatspezifikation](#) [▶ 687].

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
FUNCTION AcsiQuality_To_String : STRING(18)
VAR_INPUT
    in : ST_AcsiQuality;
END_VAR
```

Eingänge

| Name | Typ | Beschreibung |
|------|--|---|
| in | ST_AcsiQuality [▶ 463] | Die zu konvertierende Information als Quality-Struktur. |

Rückgabewert

| Name | Typ | Beschreibung |
|-----------------------|------------|--|
| AcsiQuality_To_String | STRING(18) | Konvertierte Information als Quality-String. |

Beispiel

```
METHOD AcsiQuality_ToFrom_String : BOOL
VAR_INPUT
    fbIED : REFERENCE TO FB_IED;
END_VAR
VAR
    qValue : ST_AcsiQuality;
    sValue : STRING;
END_VAR

IF NOT __ISVALIDREF(fbIED) THEN
    RETURN;
END_IF

qValue := String_To_AcsiQuality('Q#00|00000000|0|00');
sValue := AcsiQuality_To_String(qValue); (* sValue := 'Q#00|00000000|0|00' *)
IF fbIED.Relay.LLN0.Beh.q.sValue <> AcsiQuality_To_String(qValue) THEN
    qValue := fbIED.Relay.LLN0.Beh.q.qValue;
END_IF

qValue := String_To_AcsiQuality('Q#01|00110011|0|01');
sValue := AcsiQuality_To_String(qValue); (* sValue := 'Q#01|00110011|0|01' *)
IF fbIED.Relay.LLN0.Health.q.sValue <> AcsiQuality_To_String(qValue) THEN
    qValue := fbIED.Relay.LLN0.Health.q.qValue;
END_IF

qValue := String_To_AcsiQuality('Q#10|11001100|1|10');
sValue := AcsiQuality_To_String(qValue); (* sValue := 'Q#10|11001100|1|10' *)
IF fbIED.Relay.LLN0.Mod_.q.sValue <> AcsiQuality_To_String(qValue) THEN
    qValue := fbIED.Relay.LLN0.Mod_.q.qValue;
END_IF

AcsiQuality_ToFrom_String := TRUE;
```

6.4.13 AcsiQuality_To_WORD

Konvertiert die Komponenten des strukturierten Typs [ST_AcsiQuality](#) [▶ 463] in Binärdaten des Typs WORD. Die Kodierung der Komponenten im WORD entspricht der TwinCAT [Qualiy-WORD-Formatspezifikation](#) [▶ 688].

Namensraum: [Tc3_Acsi \[► 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
FUNCTION AcsiQuality_To_Word : WORD(0..8191)
VAR_INPUT
    in : ST_AcsiQuality;
END_VAR
```

Eingänge

| Name | Typ | Beschreibung |
|------|---|---|
| in | ST_AcsiQuality
[► 463] | Die zu konvertierende Information Quality-Struktur. |

Rückgabewert

| Name | Typ | Beschreibung |
|---------------------|---------------|--|
| AcsiQuality_To_WORD | WORD(0..8191) | Konvertierte Information als Quality-Binärdaten. |

Beispiel

```
METHOD AcsiQuality_ToFrom_WORD : BOOL
VAR_INPUT
    fbIED : REFERENCE TO FB_IED;
END_VAR
VAR
    qValue : ST_AcsiQuality;
    sValue : STRING;
    nValue : WORD;
END_VAR

IF NOT __ISVALIDREF(fbIED) THEN
    RETURN;
END_IF

qValue := WORD_To_AcsiQuality(0);
sValue := AcsiQuality_To_String(qValue); (* sValue := 'Q#00|00000000|0|00' *)
nValue := AcsiQuality_To_Word(qValue); (* nValue := 2#00000000_00000000 *)
IF fbIED.Relay.LLN0.Beh.q.nValue <> nValue THEN
    qValue := fbIED.Relay.LLN0.Beh.q.qValue;
END_IF

qValue := String_To_AcsiQuality('Q#01|00110011|0|01');
sValue := AcsiQuality_To_String(qValue); (* sValue := 'Q#01|00110011|0|01' *)
nValue := AcsiQuality_To_Word(qValue); (* nValue := 2#00010011_00110010 *)
IF fbIED.Relay.LLN0.Health.q.nValue <> nValue THEN
    qValue := fbIED.Relay.LLN0.Health.q.qValue;
END_IF

qValue := String_To_AcsiQuality('Q#10|11001100|1|10');
sValue := AcsiQuality_To_String(qValue); (* sValue := 'Q#10|11001100|1|10' *)
nValue := AcsiQuality_To_Word(qValue); (* nValue := 2#00001100_11001101 *)
IF fbIED.Relay.LLN0.Mod.q.nValue <> nValue THEN
    qValue := fbIED.Relay.LLN0.Mod.q.qValue;
END_IF

AcsiQuality_ToFrom_WORD := TRUE;
```

6.4.14 AcsiQualitySource_To_Source

Konvertiert die Source-Bit-Komponente des strukturierten Typs: [ST_AcsiQuality \[► 463\]](#) in den Aufzählungstyp: [E_AcsiQualitySource \[► 424\]](#). Es wird nur die Source Bit-Komponente konvertiert.

Namensraum: [Tc3_Acsi \[► 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
FUNCTION AcsiQualitySource_To_Source : E_AcsiQualitySource
VAR_INPUT
    in : ST_AcsiQuality;
END_VAR
```

 **Eingänge**

| Name | Typ | Beschreibung |
|------|---|--|
| in | ST_AcsiQuality
[▶ 463] | Die zu konvertierende Information als Quality-Source-Bit-Komponente. |

 **Rückgabewert**

| Name | Typ | Beschreibung |
|-----------------------------|---|---|
| AcsiQualitySource_To_Source | E_AcsiQualitySourc
e [▶ 424] | Konvertierte Information als Quality-Source-Aufzählungstyp. |

Beispiel

```
METHOD AcsiQualitySource_ToFrom_Source : BOOL
VAR_INPUT
    fbIED : REFERENCE TO FB_IED;
END_VAR
VAR
    qValue : ST_AcsiQuality;
    sValue : STRING;
END_VAR

IF NOT __ISVALIDREF(fbIED) THEN
    RETURN;
END_IF

Validity_To_AcsiQualityValidity(E_AcsiQualityValidity.Good, bValidity0=>qValue.Validity0, bValidity1
=> qValue.Validity1 );
qValue.Overflow := FALSE;
qValue.OutOfRange := FALSE;
qValue.BadReference := FALSE;
qValue.Oscillatory := FALSE;
qValue.Failure := FALSE;
qValue.OldData := FALSE;
qValue.Inconsistent := FALSE;
qValue.Inaccurate := FALSE;
qValue.Source := Source_To_AcsiQualitySource(E_AcsiQualitySource.Substituted);
qValue.Test := TRUE;
qValue.OperatorBlocked := FALSE;
sValue := AcsiQuality_To_String(qValue); (* sValue := 'Q#00|00000000|1|10' *)

IF fbIED.Relay.LLN0.Beh.q.eSource <> AcsiQualitySource_To_Source(qValue) THEN
    qValue := fbIED.Relay.LLN0.Beh.q.qValue;
END_IF

AcsiQualitySource_ToFrom_Source := TRUE;
```

6.4.15 AcsiQualityValidity_To_Validity

Konvertiert die Validity0- und Validity1-Bit-Komponenten des strukturierten Typs: [ST_AcsiQuality \[▶ 463\]](#) in den Aufzählungstyp: [E_AcsiQualityValidity \[▶ 425\]](#). Es werden nur die Validity0- und Validity1-Bit-Komponenten konvertiert. Siehe auch: [Kodierung der Quality-Validity0..1-Bits \[▶ 687\]](#).

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
FUNCTION AcsiQualityValidity_To_Validity : E_AcsiQualityValidity
VAR_INPUT
    in : ST_AcsiQuality;
END_VAR
```

Eingänge

| Name | Typ | Beschreibung |
|------|--|---|
| in | ST_AcsiQuality [▶ 463] | Die zu konvertierende Information als Quality-Validity-Bit-Komponenten. |

Rückgabewert

| Name | Typ | Beschreibung |
|---------------------------------|---|---|
| AcsiQualityValidity_To_Validity | E_AcsiQualityValidity [▶ 425] | Konvertierte Information als Quality-Validity-Aufzählungstyp. |

Beispiel

```
METHOD AcsiQualityValidity_ToFrom_Validity : BOOL
VAR_INPUT
    fbIED : REFERENCE TO FB_IED;
END_VAR
VAR
    qValue : ST_AcsiQuality;
    sValue : STRING;
END_VAR

IF NOT __ISVALIDREF(fbIED) THEN
    RETURN;
END_IF
Validity_To_AcsiQualityValidity(E_AcsiQualityValidity.Good, bValidity0=>qValue.Validity0, bValidity1=>qValue.Validity1);
qValue.Overflow := FALSE;
qValue.OutOfRange := FALSE;
qValue.BadReference := FALSE;
qValue.Oscillatory := FALSE;
qValue.Failure := FALSE;
qValue.OldData := FALSE;
qValue.Inconsistent := FALSE;
qValue.Inaccurate := FALSE;
qValue.Source := Source_To_AcsiQualitySource(E_AcsiQualitySource.Process);
qValue.Test := FALSE;
qValue.OperatorBlocked := FALSE;
sValue := AcsiQuality_To_String(qValue); (* sValue := 'Q#00|00000000|0|00' *)

IF fbIED.Relay.LLN0.Beh.q.eValidity <> AcsiQualityValidity_To_Validity(qValue) THEN
    qValue := fbIED.Relay.LLN0.Beh.q.qValue;
END_IF

AcsiQualityValidity_ToFrom_Validity := TRUE;
```

6.4.16 AcsiRangeConfig

Initialisiert die Datenelemente der Struktur: ST_AcsiRangeConfig.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

```
FUNCTION AcsiRangeConfig : ST_AcsiRangeConfig
VAR_INPUT
    hhLim : ST_AcsiAnalogueValue;
    hLim : ST_AcsiAnalogueValue;
    lLim : ST_AcsiAnalogueValue;
    llLim : ST_AcsiAnalogueValue;
    min_ : ST_AcsiAnalogueValue;
```

```

max_ : ST_AcsiAnalogueValue;
limDb : DWORD;
END_VAR

```

 **Eingänge**

| Name | Typ | Beschreibung |
|-------|--|--|
| hhLim | ST_AcsiAnalogueValue [▶ 455] | Datenelement „hhLim“, das gesetzt werden soll. |
| hLim | ST_AcsiAnalogueValue [▶ 455] | Datenelement „hLim“, das gesetzt werden soll. |
| lLim | ST_AcsiAnalogueValue [▶ 455] | Datenelement „lLim“, das gesetzt werden soll. |
| llLim | ST_AcsiAnalogueValue [▶ 455] | Datenelement „llLim“, das gesetzt werden soll. |
| min_ | ST_AcsiAnalogueValue [▶ 455] | Datenelement „min_“, das gesetzt werden soll. |
| max_ | ST_AcsiAnalogueValue [▶ 455] | Datenelement „max_“, das gesetzt werden soll. |
| limDb | DWORD | Datenelement „limDb“, das gesetzt werden soll. |

 **Rückgabewert**

| Name | Typ | Beschreibung |
|-----------------|--|--|
| AcsiRangeConfig | ST_AcsiRangeConfig [▶ 464] | Struktur mit initialisierten Datenelementen. |

Beispiel

```

METHOD FINAL Sample_AcsiRangeConfig : BOOL
VAR
    range: ST_AcsiRangeConfig;
END_VAR

range:=AcsiRangeConfig(hhLim:=AcsiAnalogueValue(0,0), hLim:=AcsiAnalogueValue(0,0), lLim:=AcsiAnalogueValue(0,0), llLim:=AcsiAnalogueValue(0,0), min_:=AcsiAnalogueValue(0,0), max_:=AcsiAnalogueValue(0,0), limDb:=0);
Sample_AcsiRangeConfig:=TRUE;

```

6.4.17 AcsiReasonCode

Initialisiert die Datenelemente der Struktur: ST_AcsiReasonCode.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

```

FUNCTION AcsiReasonCode : ST_AcsiReasonCode
VAR_INPUT
    DataChange          : BOOL;
    QualityChange       : BOOL;
    DataUpdate          : BOOL;
    Integrity            : BOOL;
    GeneralInterrogation : BOOL;
    ApplicationTrigger   : BOOL;
END_VAR

```

 **Eingänge**

| Name | Typ | Beschreibung |
|----------------------|------|---|
| DataChange | BOOL | Datenelement „DataChange“, das gesetzt werden soll. |
| QualityChange | BOOL | Datenelement „QualityChange“, das gesetzt werden soll. |
| DataUpdate | BOOL | Datenelement „DataUpdate“, das gesetzt werden soll. |
| Integrity | BOOL | Datenelement „Integrity“, das gesetzt werden soll. |
| GeneralInterrogation | BOOL | Datenelement „GeneralInterrogation“, das gesetzt werden soll. |
| ApplicationTrigger | BOOL | Datenelement „ApplicationTrigger“, das gesetzt werden soll. |

Rückgabewert

| Name | Typ | Beschreibung |
|----------------|--|--|
| AcsiReasonCode | ST_AcsiReasonCode ▶ 465 | Struktur mit initialisierten Datenelementen. |

Beispiel

```
METHOD FINAL Sample_AcsiReasonCode : BOOL
VAR
    reason : ST_AcsiReasonCode;
END_VAR

reason:=AcsiReasonCode(DataChange:=TRUE, QualityChange:=TRUE, DataUpdate:=TRUE,
    Integrity:=FALSE, GeneralInterrogation:=FALSE, ApplicationTrigger:=FALSE);
Sample_AcsiReasonCode:=TRUE;
```

6.4.18 AcsiScaledValueConfig

Initialisiert die Datenelemente der Struktur: [ST_AcsiScaledValueConfig](#).

Namensraum: [Tc3_Acsi](#) |▶ 96|

Bibliothek: [Tc3_Acsi](#) ([Tc3_Acsi.compiled-library](#))

```
FUNCTION AcsiScaledValueConfig : ST_AcsiScaledValueConfig
VAR_INPUT
    scaleFactor : REAL;
    offset      : REAL;
END_VAR
```

Eingänge

| Name | Typ | Beschreibung |
|-------------|------|--|
| scaleFactor | REAL | Datenelement „scaleFactor“, das gesetzt werden soll. |
| offset | REAL | Datenelement „offset“, das gesetzt werden soll. |

Rückgabewert

| Name | Typ | Beschreibung |
|-----------------------|---|--|
| AcsiScaledValueConfig | ST_AcsiScaledValueConfig ▶ 465 | Struktur mit initialisierten Datenelementen. |

Beispiel

```
METHOD FINAL Sample_AcsiScaledValueConfig : BOOL
VAR
    config: ST_AcsiScaledValueConfig;
END_VAR

config:=AcsiScaledValueConfig(scaleFactor:=1.0, offset:=100);
Sample_AcsiScaledValueConfig:=TRUE;
```

6.4.19 AcsiSvOptionalFields

Initialisiert die Datenelemente der Struktur: [ST_AcsiSvOptionalFields](#).

Namensraum: [Tc3_Acsi](#) |▶ 96|

Bibliothek: [Tc3_Acsi](#) ([Tc3_Acsi.compiled-library](#))

```
FUNCTION AcsiSvOptionalFields : ST_AcsiSvOptionalFields
VAR_INPUT
    RefreshTime      : BOOL;
    SampleSynchronized : BOOL;
    SampleRate       : BOOL;
    DataSetName      : BOOL;
    Security          : BOOL;
```



```

SampleMode      : BOOL;
SynchSourceIdentity : BOOL;
END_VAR

```

 **Eingänge**

| Name | Typ | Beschreibung |
|---------------------|------|--|
| RefreshTime | BOOL | Datenelement „RefreshTime“, das gesetzt werden soll. |
| SampleSynchronized | BOOL | Datenelement „SampleSynchronized“, das gesetzt werden soll. |
| SampleRate | BOOL | Datenelement „SampleRate“, das gesetzt werden soll. |
| DataSetName | BOOL | Datenelement „DataSetName“, das gesetzt werden soll. |
| Security | BOOL | Datenelement „Security“, das gesetzt werden soll. |
| SampleMode | BOOL | Datenelement „SampleMode“, das gesetzt werden soll. |
| SynchSourceIdentity | BOOL | Datenelement „SynchSourceIdentity“, das gesetzt werden soll. |

 **Rückgabewert**

| Name | Typ | Beschreibung |
|----------------------|--|--|
| AcsiSvOptionalFields | ST_AcsiSvOptionalFields
▶ 466 | Struktur mit initialisierten Datenelementen. |

Beispiel

```

METHOD FINAL Sample_AcsiSvOptionalFields : BOOL
VAR
    fields : ST_AcsiSvOptionalFields;
END_VAR

fields:=AcSiSvOptionalFields(RefreshTime:=FALSE, SampleSynchronized:=FALSE,
    SampleRate:=FALSE, DataSetName:=TRUE, Security:=FALSE,
    SampleMode:=FALSE, SynchSourceIdentity:=FALSE);
Sample_AcsiSvOptionalFields:=TRUE;

```

6.4.20 AcSiTriggerConditions

Initialisiert die Datenelemente der Struktur: ST_AcsiTriggerConditions.

Namensraum: [Tc3_Acsi](#) [▶ 96](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

```

FUNCTION AcSiTriggerConditions : ST_AcsiTriggerConditions
VAR_INPUT
    DataChange      : BOOL;
    QualityChange   : BOOL;
    DataUpdate      : BOOL;
    Integrity        : BOOL;
    GeneralInterrogation : BOOL;
END_VAR

```

 **Eingänge**

| Name | Typ | Beschreibung |
|----------------------|------|---|
| DataChange | BOOL | Datenelement „DataChange“, das gesetzt werden soll. |
| QualityChange | BOOL | Datenelement „QualityChange“, das gesetzt werden soll. |
| DataUpdate | BOOL | Datenelement „DataUpdate“, das gesetzt werden soll. |
| Integrity | BOOL | Datenelement „Integrity“, das gesetzt werden soll. |
| GeneralInterrogation | BOOL | Datenelement „GeneralInterrogation“, das gesetzt werden soll. |

Rückgabewert

| Name | Typ | Beschreibung |
|-----------------------|--|--|
| AcsiTriggerConditions | ST_AcsiTriggerConditions [▶ 466] | Struktur mit initialisierten Datenelementen. |

Beispiel

```
METHOD FINAL Sample_AcsiTriggerConditions : BOOL
VAR
  trigger : ST_AcsiTriggerConditions;
END_VAR
```

```
trigger:=AcsiTriggerConditions(DataChange:=TRUE, QualityChange:=TRUE, DataUpdate:=TRUE, Integrity:=FALSE, GeneralInterrogation:=FALSE);
Sample_AcsiTriggerConditions:=TRUE;
```

6.4.21 AcsiUnit

Initialisiert die Datenelemente der Struktur: ST_AcsiUnit.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

```
FUNCTION AcsiUnit : ST_AcsiUnit
VAR_INPUT
  SIUnit : E_AcsiSIUnit;
  multiplier : E_AcsiMultiplier;
END_VAR
```

Eingänge

| Name | Typ | Beschreibung |
|------------|--|---|
| SIUnit | E_AcsiSIUnit [▶ 432] | Datenelement „SIUnit“, das gesetzt werden soll. |
| multiplier | E_AcsiMultiplier [▶ 416] | Datenelement „multiplier“, das gesetzt werden soll. |

Rückgabewert

| Name | Typ | Beschreibung |
|----------|---|--|
| AcsiUnit | ST_AcsiUnit [▶ 467] | Struktur mit initialisierten Datenelementen. |

Beispiel

```
METHOD FINAL Sample_AcsiUnit : BOOL
VAR
  unit: ST_AcsiUnit;
END_VAR
```

```
unit:=AcsiUnit(SIUnit:=E_AcsiSIUnit.Ampere, multiplier:=E_AcsiMultiplier.Milli);
Sample_AcsiUnit:=TRUE;
```

6.4.22 AcsiValWithTrans

Initialisiert die Datenelemente der Struktur: ST_AcsiValWithTrans.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

```
FUNCTION AcsiValWithTrans : ST_AcsiValWithTrans
VAR_INPUT
  posVal : SINT;
  transInd : BOOL;
END_VAR
```

 **Eingänge**

| Name | Typ | Beschreibung |
|----------|------|---|
| posVal | SINT | Datenelement „posVal“, das gesetzt werden soll. |
| transInd | BOOL | Datenelement „transInd“, das gesetzt werden soll. |

 **Rückgabewert**

| Name | Typ | Beschreibung |
|------------------|---|--|
| AcsiValWithTrans | ST_AcsiValWithTrans [▶ 467] | Struktur mit initialisierten Datenelementen. |

Beispiel

```
METHOD FINAL Sample_AcsiValWithTrans : BOOL
VAR
    valT : ST_AcsiValWithTrans;
END_VAR

valT:=AcsiValWithTrans(posVal:=1, transInd:=FALSE);
Sample_AcsiValWithTrans:=TRUE;
```

6.4.23 AcsiVector

Initialisiert die Datenelemente der Struktur: [ST_AcsiVector](#).

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: [Tc3_Acsi](#) (Tc3_Acsi.compiled-library)

```
FUNCTION AcsiVector : ST_AcsiVector
VAR_INPUT
    mag : ST_AcsiAnalogueValue;
    ang : ST_AcsiAnalogueValue;
END_VAR
```

 **Eingänge**

| Name | Typ | Beschreibung |
|------|--|--|
| mag | ST_AcsiAnalogueValue [▶ 455] | Datenelement „mag“, das gesetzt werden soll. |
| ang | ST_AcsiAnalogueValue [▶ 455] | Datenelement „ang“, das gesetzt werden soll. |

 **Rückgabewert**

| Name | Typ | Beschreibung |
|------------|---------------------------------------|--|
| AcsiVector | ST_AcsiVector [▶ 468] | Struktur mit initialisierten Datenelementen. |

Beispiel

```
METHOD FINAL Sample_AcsiVector : BOOL
VAR
    vector : ST_AcsiVector;
END_VAR

vector:=AcsiVector(mag:=AcsiAnalogueValue(0, 0), ang:=AcsiAnalogueValue(0, 0));
Sample_AcsiVector:=TRUE;
```

6.4.24 AnyBuffer_To_OctetString

Konvertiert und kopiert die Daten einer beliebigen Variablen in den Oktett-String oder ein eindimensionales Byte-Array.

Namensraum: [Tc3_Collections](#) [▶ 96]

Bibliothek: [Tc3_Collections](#) (Tc3_Collections.compiled-library)

```

FUNCTION AnyBuffer_To_OctetString : UDINT
VAR_IN_OUT
  target      : ARRAY[*] OF BYTE;
END_VAR
VAR_INPUT
  pSource     : PVOID;
  cbSource    : UDINT;
  bSwapBytes  : BOOL;
END_VAR
VAR_OUTPUT
  bUnderflow  : BOOL;
  bOverflow   : BOOL;
END_VAR

```

/ Ein-/Ausgänge

| Name | Typ | Beschreibung |
|--------|------------------|---------------|
| target | ARRAY[*] OF BYTE | Zielvariable. |

Eingänge

| Name | Typ | Beschreibung |
|------------|-------|---|
| pSource | PVOID | Adresszeiger der Quellvariablen. |
| cbSource | UDINT | Bytegröße des Quellvariablen. |
| bSwapBytes | BOOL | Bei TRUE werden die Datenbytes getauscht. |

Ausgänge

| Name | Typ | Beschreibung |
|------------|------|--|
| bUnderflow | BOOL | Wenn TRUE dann ist die Quellvariable kleiner als die Zielvariable. |
| bOverflow | BOOL | Wenn TRUE dann ist die Quellvariable größer als die Zielvariable. |

Rückgabewert

| Name | Typ | Beschreibung |
|--------------------------|-------|--|
| AnyBuffer_To_OctetString | UDINT | Anzahl der erfolgreich kopierten Datenbytes. |

Beispiel

```

METHOD FINAL Sample_AnyBuffer_To_OctetString : BOOL
VAR
  sValue      : STRING:='0123456789';
  ui64        : LWORD:=16#3031323334353637;
  ui32        : UDINT:=16#30313233;
  o6          : T_OCTET6;
  o8          : T_OCTET8;
  o64         : T_OCTET64;
  size        : UDINT;
  bOverflow   : BOOL;
  bUnderflow  : BOOL;
END_VAR
size:=AnyBuffer_To_OctetString(target:=o6, pSource:=ADR(sValue), cbSource:=SIZEOF(sValue), bSwapByte
s:=FALSE, bUnderflow=>bUnderflow, bOverflow=>bOverflow);(* size:=6 *)
size:=AnyBuffer_To_OctetString(target:=o8, pSource:=ADR(ui64), cbSource:=SIZEOF(ui64), bSwapBytes:=T
RUE, bUnderflow=>bUnderflow, bOverflow=>bOverflow);(* size:=8 *)
size:=AnyBuffer_To_OctetString(target:=o64, pSource:=ADR(ui32), cbSource:=SIZEOF(ui32), bSwapBytes:=
TRUE, bUnderflow=>bUnderflow, bOverflow=>bOverflow);(* size:=4 *)
Sample_AnyBuffer_To_OctetString:=TRUE;

```

6.4.25 BinaryTime

Initialisiert die Datenelemente der Struktur: T_BinaryTime.

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

```
FUNCTION BinaryTime : T_BinaryTime
VAR_INPUT
    timeOfDay : TOD;
    day       : WORD;
END_VAR
```

Eingänge

| Name | Typ | Beschreibung |
|-----------|------|--|
| timeOfDay | TOD | Anzahl der vergangenen Millisekunden seit Mitternacht des aktuellen Tages. |
| day | WORD | Anzahl der vergangenen Tage seit dem 1. Januar 1984. |

Rückgabewert

| Name | Typ | Beschreibung |
|------------|--|--|
| BinaryTime | T_BinaryTime [▶ 476] | Struktur mit initialisierten Datenelementen. |

Beispiel

```
METHOD FINAL Sample_BinaryTime : BOOL
VAR
    t: T_BinaryTime;
END_VAR

t:=BinaryTime(timeOfDay:=TOD#12:00:00.000, day:=DATE_To_BinaryTime6Day(in:=D#2023-01-01));
Sample_BinaryTime:=(t.timeOfDay = TOD#12:00:00.000) AND (t.day = 14245);
```

6.4.26 BinaryTime_To_SystemTime

Konvertiert die Zeitinformation des Typs [T_BinaryTime](#) [[▶ 476](#)] (Binary-Time-Format) in die Zeitinformation des Typs [TIMESTRUCT](#) [[▶ 454](#)] (System-Time-Format). Beim Konvertierungsfehler hat die Ausgangsvariable „bOverflow“ den Wert „TRUE“ und der zurückgelieferte System-Time-Wert das Datum:1970-01-01 und die Uhrzeit: 0h0m0s0ms.

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
FUNCTION BinaryTime_To_SystemTime : TIMESTRUCT
VAR_INPUT
    in : T_BinaryTime;
END_VAR
VAR_OUTPUT
    bOverflow : BOOL;
END_VAR
```

Eingänge

| Name | Typ | Beschreibung |
|------|--|---|
| in | T_BinaryTime [▶ 476] | Die zu konvertierende Zeitinformation als Binary-Time-Struktur. |

Ausgänge

| Name | Typ | Beschreibung |
|-----------|------|---|
| bOverflow | BOOL | Datumsüberlauf wenn "TRUE". In diesem Fall hat die zu konvertierende Binary-Time ein Datum (T_BinaryTime [▶ 476].day member) welches größer ist als 2106-02-07. |

Rückgabewert

| Name | Typ | Beschreibung |
|--------------------------|--------------------|--|
| BinaryTime_To_SystemTime | TIMESTRUCT [▶ 454] | Konvertierte Zeitinformation als System-Time-Struktur. |

Beispiel

```
PROGRAM MAIN
VAR
  tSystem   : TIMESTRUCT;
  tBinary   : T_BinaryTime;
  bOverflow : BOOL;
END_VAR

tSystem.wYear:=2022;
tSystem.wMonth:=9;
tSystem.wDay:=16;
tSystem.wHour:=12;
tSystem.wMinute:=52;
tSystem.wSecond:=4;
tSystem.wMilliseconds:=500;
tBinary:=SystemTime_To_BinaryTime(in:=tSystem, bOverflow=>bOverflow);
(* Returns tBinary.timeOfDay = TOD#12:52:04.500 and tBinary.day = 14138 *)

tBinary.day:=1;
tBinary.timeOfDay:=TOD#01:30:15.500;
tSystem:=BinaryTime_To_SystemTime(in:=tBinary, bOverflow=>bOverflow);(* Returns system-
time date: 1984-01-02 and time: 01h30m15s500ms*)
```

6.4.27 BinaryTime6_To_LWORD

Konvertiert die Komponenten des strukturierten Typs [T_BinaryTime \[▶ 476\]](#) in die Binärdaten des Typs LWORD. Die Kodierung der Komponenten im LWORD entspricht der TwinCAT [Binary-Time LWORD-Formatspezifikation \[▶ 681\]](#).

Namensraum: [Tc3_Collections \[▶ 96\]](#)

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
FUNCTION BinaryTime6_To_LWORD : LWORD(0.. 16#FFFFFFFFFFFFFF);
VAR_INPUT
  in      : T_BinaryTime;
END_VAR
VAR_OUTPUT
  bError : BOOL;
END_VAR
```

Eingänge

| Name | Typ | Beschreibung |
|------|----------------------|---|
| in | T_BinaryTime [▶ 476] | Die zu konvertierende Zeitinformation als Binary-Time Struktur. |

 **Ausgänge**

| Name | Typ | Beschreibung |
|--------|------|--|
| bError | BOOL | Der Ausgang ist „TRUE“ wenn der Wert der in.timeOfDay-Komponente größer ist als „TOD#23:59:59.999“. In diesem Fall liefert die Funktion den Rückgabewert „0“ zurück. |

 **Rückgabewert**

| Name | Typ | Beschreibung |
|---------------------|----------------------------|--|
| BinaryTime_To_LWORD | LWORD(0..16#FFFFFFFFFFFF); | Konvertierte Zeitinformation als Binary-Time-Binärdaten. |

Beispiel

```

METHOD BinaryTime6_ToFrom_LWORD : BOOL
VAR_INPUT
    fbIED : REFERENCE TO FB_IED;
END_VAR
VAR
    tValue : T_BinaryTime;
    sValue : STRING;
END_VAR

IF NOT __ISVALIDREF(fbIED) THEN
    RETURN;
END_IF

tValue := LWORD_To_BinaryTime6(0);
sValue := BinaryTime6_To_String(tValue); (* sValue := 'BT#1984-01-01-00:00:00' *)
IF fbIED.Relay.LLN0.brcb1.TimeOfEntry.nValue <> BinaryTime6_To_LWORD(tValue) THEN
    tValue := fbIED.Relay.LLN0.brcb1.TimeOfEntry.tValue;
END_IF

tValue := LWORD_To_BinaryTime6(1);
sValue := BinaryTime6_To_String(tValue); (* sValue := 'BT#1984-01-01-00:00:00.001' *)
IF fbIED.Relay.LLN0.brcb2.TimeOfEntry.nValue <> BinaryTime6_To_LWORD(tValue) THEN
    tValue := fbIED.Relay.LLN0.brcb2.TimeOfEntry.tValue;
END_IF

tValue := String_To_BinaryTime6('BT#2017-03-21-11:22:33.44');
sValue := BinaryTime6_To_String(tValue); (* sValue := 'BT#2017-03-21-11:22:33.440' *)
IF fbIED.Relay.LLN0.brcb3.TimeOfEntry.nValue <> BinaryTime6_To_LWORD(tValue) THEN
    tValue := fbIED.Relay.LLN0.brcb3.TimeOfEntry.tValue;
END_IF

BinaryTime6_ToFrom_LWORD := TRUE;
    
```

6.4.28 BinaryTime6_To_String

Konvertiert die Komponenten des strukturierten Typs T_BinaryTime [[▶ 476](#)] in einen formatierten Binary-Time-String. Die Zeichenfolge des resultierenden Strings hat folgenden Aufbau: 'BT#[YYYY-MM-DD-]hh:mm:ss[.nnn]'. Sie entspricht der TwinCAT Binary-Time-String-Formatspezifikation [[▶ 681](#)].

Namensraum: Tc3_Collections [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```

FUNCTION BinaryTime6_To_String : STRING(26)
VAR_INPUT
    in : T_BinaryTime;
END_VAR
VAR_OUTPUT
    bError : BOOL;
END_VAR
    
```

Eingänge

| Name | Typ | Beschreibung |
|------|--|---|
| in | T_BinaryTime [▶ 476] | Die zu konvertierende Zeitinformation als Binary-Time-Struktur. |

Ausgänge

| Name | Typ | Beschreibung |
|--------|------|--|
| bError | BOOL | Der Ausgang ist „TRUE“ wenn der Wert der in.day-Komponente größer ist als „16#AE35“ (dies entspricht dem 7. Februar 2106). In diesem Fall liefert die Funktion als Rückgabewert einen Leerstring zurück. |

Rückgabewert

| Name | Typ | Beschreibung |
|-----------------------|------------|--|
| BinaryTime6_To_String | STRING(26) | Konvertierte Zeitinformation als Binary-Time-String. |

Beispiel

```

METHOD BinaryTime6_ToFrom_String : BOOL
VAR_INPUT
    fbIED : REFERENCE TO FB_IED;
END_VAR
VAR
    tValue : T_BinaryTime;
    sValue : STRING;
END_VAR

IF NOT __ISVALIDREF(fbIED) THEN
    RETURN;
END_IF

tValue := String_To_BinaryTime6('BT#01:02:03');
sValue := BinaryTime6_To_String(tValue); (* sValue := 'BT#1984-01-01-01:02:03' *)
IF fbIED.Relay.LLN0.brcb1.TimeOfEntry.sValue <> sValue THEN
    tValue := fbIED.Relay.LLN0.brcb1.TimeOfEntry.tValue;
END_IF

tValue := String_To_BinaryTime6('BT#04:05:06.7');
sValue := BinaryTime6_To_String(tValue); (* sValue := 'BT#1984-01-01-04:05:06.700' *)
IF fbIED.Relay.LLN0.brcb2.TimeOfEntry.sValue <> sValue THEN
    tValue := fbIED.Relay.LLN0.brcb2.TimeOfEntry.tValue;
END_IF

tValue := String_To_BinaryTime6('BT#2017-01-18-11:12:13');
sValue := BinaryTime6_To_String(tValue); (* sValue := 'BT#2017-01-18-11:12:13' *)
IF fbIED.Relay.LLN0.brcb3.TimeOfEntry.sValue <> sValue THEN
    tValue := fbIED.Relay.LLN0.brcb3.TimeOfEntry.tValue;
END_IF

BinaryTime6_ToFrom_String := TRUE;

```

6.4.29 BinaryTime6Day_To_Date

Konvertiert die Anzahl der vergangenen Tage seit dem 1. Januar 1984 des Typs WORD in ein Datum des Typs DATE. Die Funktion liefert den max. darstellbaren DATE-Wert „D#2106-02-07“ zurück, wenn der zu konvertierende Tag größer als der Wert „16#AE35“ ist.

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```

FUNCTION BinaryTime6Day_To_Date : DATE
VAR_INPUT
    in : WORD(0..16#AE35);
END_VAR

```



```
VAR_OUTPUT
  bError : BOOL;
END_VAR
```

 **Eingänge**

| Name | Typ | Beschreibung |
|------|------------------|--|
| in | WORD(0..16#AE35) | Die zu konvertierende Anzahl der vergangenen Tage seit dem 1. Januar 1984. Dieser Wert entspricht dem Wert der <u>T_BinaryTime</u> [▶ 476].day-Komponente. |

 **Ausgänge**

| Name | Typ | Beschreibung |
|--------|------|---|
| bError | BOOL | Der Ausgang ist „TRUE“ wenn der Eingangswert größer ist als „16#AE35“ (7 Februar 2106). In diesem Fall liefert die Funktion den Rückgabewert „D#1970-01-01“ zurück. |

 **Rückgabewert**

| Name | Typ | Beschreibung |
|------------------------|------|-----------------------|
| BinaryTime6Day_To_Date | DATE | Datum im DATE-Format. |

Beispiel

```
METHOD FINAL BinaryTime6Day_ToFrom_Date : BOOL
VAR
  tValue : T_BinaryTime;
  d      : DATE;
  sValue : STRING;
END_VAR
```

```
tValue.timeOfDay:=TOD#00:00:00.125;
tValue.day:=Date_To_BinaryTime6Day(D#1984-01-01); (* tValue.day:=16#0000 *)
sValue:=BinaryTime6_To_String(tValue); (* sValue:='BT#1984-01-01-00:00:00.125' *)
d:=BinaryTime6Day_To_Date(tValue.day); (* d:=D#1984-01-01 *)

tValue.timeOfDay:=TOD#01:02:03.125;
tValue.day:=Date_To_BinaryTime6Day(D#1984-01-02); (* tValue.day:=16#0001 *)
sValue:=BinaryTime6_To_String(tValue); (* sValue:='BT#1984-01-02-01:02:03.125' *)
d:=BinaryTime6Day_To_Date(tValue.day); (* d:=D#1984-01-02 *)

tValue.timeOfDay:=TOD#04:05:06.125;
tValue.day:=Date_To_BinaryTime6Day(D#2018-03-21); (* tValue.day:=16#30D2 *)
sValue:=BinaryTime6_To_String(tValue); (* sValue:='BT#2018-03-21-04:05:06.125' *)
d:=BinaryTime6Day_To_Date(tValue.day); (* d:=D#2018-03-21 *)

tValue.timeOfDay:=TOD#07:08:09;
tValue.day:=tValue.day + 7; (* add 7 days *)
sValue:=BinaryTime6_To_String(tValue); (* sValue:='BT#2018-03-28-07:08:09' *)
d:=BinaryTime6Day_To_Date(tValue.day); (* d:=D#2018-03-28 *)

BinaryTime6Day_ToFrom_Date:=TRUE;
```

6.4.30 BYTE_To_UtcTimeQuality

Konvertiert die Binärdaten des Typs BYTE in die Komponenten des strukturierten Typs T_UtcTimeQuality [[▶ 480](#)]. Die Kodierung der Komponenten im BYTE entspricht der TwinCAT UTC-Time-Quality-BYTE-Formatspezifikation [[▶ 684](#)].

Namensraum: Tc3_Collections [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
FUNCTION BYTE_To_UtcTimeQuality : T_UtcTimeQuality
VAR_INPUT
    in      : BYTE;
END_VAR
```

Eingänge

| Name | Typ | Beschreibung |
|------|------|--|
| in | BYTE | Die zu konvertierende Information als UTC-Time-Quality-Binärdaten. |

Rückgabewert

| Name | Typ | Beschreibung |
|------------------------|---|---|
| BYTE_To_UtcTimeQuality | T_UtcTimeQuality
▶ 480 | Konvertierte Information als UTC-Time-Quality-Struktur. |

Beispiel

```
METHOD UtcTimeQuality_ToFrom_Byte : BOOL
VAR_INPUT
    fbIED      : REFERENCE TO FB_IED;
END_VAR
VAR
    Quality    : T_UtcTimeQuality;
    sQuality   : STRING;
    u8         : BYTE;
END_VAR

IF NOT __ISVALIDREF(fbIED) THEN
    RETURN;
END_IF

Quality := Byte_To_UtcTimeQuality(0);
u8 := UtcTimeQuality_To_Byte(Quality); (* u8 := 2#00000000 => 0 bits accuracy *)
sQuality := UtcTimeQuality_To_String(Quality); (* sQuality := 'UQ#000|00' *)
IF UtcTimeQuality_To_Byte(fbIED.Relay.LLN0.Beh.t.Quality) <> u8 THEN
    Quality := fbIED.Relay.LLN0.Beh.t.Quality;
END_IF

Quality := String_To_UtcTimeQuality('UQ#000|03');
u8 := UtcTimeQuality_To_Byte(Quality); (* u8 := 2#11000000 => 3 bits accuracy *)
sQuality := UtcTimeQuality_To_String(Quality); (* sQuality := 'UQ#000|03' *)
IF UtcTimeQuality_To_Byte(fbIED.Relay.LLN0.Health.t.Quality) <> u8 THEN
    Quality := fbIED.Relay.LLN0.Health.t.Quality;
END_IF

Quality := String_To_UtcTimeQuality('UQ#001|05');
u8 := UtcTimeQuality_To_Byte(Quality);
(* u8 := 2#10100100 => 5 bits accuracy, ClockNotSynchronized := 1 *)
sQuality := UtcTimeQuality_To_String(Quality); (* sQuality := 'UQ#001|05' *)
IF UtcTimeQuality_To_Byte(fbIED.Relay.LLN0.Mod_.t.Quality) <> u8 THEN
    Quality := fbIED.Relay.LLN0.Mod_.t.Quality;
END_IF

Quality := String_To_UtcTimeQuality('UQ#000|31');
u8 := UtcTimeQuality_To_Byte(Quality); (* u8 := 2#11111000 => unspecified accuracy *)
sQuality := UtcTimeQuality_To_String(Quality); (* sQuality := 'UQ#000|31' *)
IF UtcTimeQuality_To_Byte(fbIED.Relay.MMXU1.phV.phsA.t.Quality) <> u8 THEN
    Quality := fbIED.Relay.MMXU1.phV.phsA.t.Quality;
END_IF

UtcTimeQuality_ToFrom_Byte := TRUE;
```

6.4.31 CltsEveryBool

Diese Funktion prüft ob alle Elemente eines booleschen Arrays den gleichen Wert haben wie der bTest Funktionseingangsparameter.

Namensraum: [Tc3_Collections](#) [▶ 96]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
FUNCTION CltsEveryBool : BOOL
VAR_INPUT
    bTest: BOOL;
END_VAR
VAR_IN_OUT
    in: ARRAY[*] OF BOOL;
END_VAR
VAR_OUTPUT
    index: DINT;
END_VAR
```

Eingänge

| Name | Typ | Beschreibung |
|-------|------|---|
| bTest | BOOL | Testwert, der mit allen Arrayelementen verglichen wird. |

Ein-/Ausgänge

| Name | Typ | Beschreibung |
|------|------------------|---|
| in | ARRAY[*] OF BOOL | Die zu prüfende boolesche Array-Variable. |

Ausgänge

| Name | Typ | Beschreibung |
|-------|------|--|
| index | DINT | Indexnummer des ersten Arrayelements dessen Wert ungleich dem bTest Eingangsparmeter ist. Dieser Wert ist 0 wenn der Funktionsrückgabeparmeter „TRUE“ ist. |

Rückgabewert

| Name | Typ | Beschreibung |
|---------------|------|---|
| CltsEveryBool | BOOL | „TRUE“ wenn alle Arrayelemente den gleichen Wert haben wie der bTest Funktionseingangsparmeter. „FALSE“ wenn mindestens ein Arrayelement einen ungleichen Wert hat wie der bTest Funktionseingangsparmeter. |

Beispiel

```
PROGRAM MAIN
VAR
    a: ARRAY[1..5] OF BOOL;
    index: DINT;
    bResult: BOOL;
END_VAR

a[1]:=FALSE;
a[2]:=FALSE;
a[3]:=TRUE;
a[4]:=FALSE;
a[5]:=TRUE;
bResult:=CltsEveryBool(bTest:=TRUE, in:=a, index=>index);(* return value is FALSE, index = 1 *)
bResult:=CltsEveryBool(bTest:=FALSE, in:=a, index=>index);(* return value is FALSE, index = 3 *)
a[1]:=TRUE;
a[2]:=TRUE;
a[4]:=TRUE;
bResult:=CltsEveryBool(bTest:=TRUE, in:=a, index=>index);(* return value is TRUE, index = 0 *)
```

6.4.32 CltsSomeBool

Diese Funktion prüft ob mindestens ein Arrayelement eines booleschen Arrays den gleichen Wert hat wie der bTest Funktionseingangsparameter.

Namensraum: Tc3_Collections [▶ 96]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
FUNCTION CltsSomeBool : BOOL
VAR_INPUT
    bTest: BOOL;
END_VAR
VAR_IN_OUT
    in: ARRAY[*] OF BOOL;
END_VAR
VAR_OUTPUT
    index: DINT;
END_VAR
```

Eingänge

| Name | Typ | Beschreibung |
|-------|------|---|
| bTest | BOOL | Testwert, der mit allen Arrayelementen verglichen wird. |

Ein/Ausgänge

| Name | Typ | Beschreibung |
|------|------------------|---|
| in | ARRAY[*] OF BOOL | Die zu prüfende boolesche Array-Variable. |

Ausgänge

| Name | Typ | Beschreibung |
|-------|------|---|
| index | DINT | Index des ersten Arrayelements dessen Wert gleich dem bTest Funktionseingangsparameter ist. Dieser Wert ist 0 wenn der Funktionsrückgabewert „FALSE“ ist. |

Rückgabewert

| Name | Typ | Beschreibung |
|--------------|------|--|
| CltsSomeBool | BOOL | „TRUE“ wenn mindestens ein Arrayelement den gleichen Wert hat wie der bTest Funktionseingangsparameter.
„FALSE“ wenn keiner der Arrayelemente den gleichen Wert hat wie der bTest Funktionseingangsparameter. |

Beispiel

```
PROGRAM MAIN
VAR
    a: ARRAY[1..5] OF BOOL;
    index: DINT;
    bResult: BOOL;
END_VAR

a[1]:=FALSE;
a[2]:=FALSE;
a[3]:=FALSE;
a[4]:=FALSE;
a[5]:=FALSE;
bResult:=CltsSomeBool(bTest:=TRUE, in:=a, index=>index);(* return value is FALSE, index = 0 *)
a[1]:=FALSE;
a[2]:=FALSE;
a[3]:=TRUE;
a[4]:=FALSE;
```

```
a[5]:=TRUE;
bResult:=CltsSomeBool(bTest:=TRUE, in:=a, index=>index);(* return value is TRUE, index = 3 *)
bResult:=CltsSomeBool(bTest:=FALSE, in:=a, index=>index);(* return value is TRUE, index = 1 *)
```

6.4.33 DATE_and_TOD_To_DT

Kombiniert das Datum des Typs DATE und die Uhrzeit des Typs TIME_OF_DAY (TOD) zur Zeitinformation des Typs DATE_AND_TIME (DT).

Namensraum: Tc3_Collections [▶ 96]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
FUNCTION DATE_and_TOD_To_DT : DT
VAR_INPUT
    d          : DATE;
    t          : TOD;
END_VAR
VAR_OUTPUT
    wMilliseconds : WORD;
END_VAR
```

Eingänge

| Name | Typ | Beschreibung |
|------|------|--------------|
| d | DATE | Datum. |
| t | TOD | Uhrzeit. |

Ausgänge

| Name | Typ | Beschreibung |
|---------------|------|---|
| wMilliseconds | WORD | Millisekunden Rest der TOD-Uhrzeit, welches nicht in den Typ DATE_AND_TIME konvertiert werden konnte. Der DATE_AND_TIME Typ hat die max. Auflösung von einer Sekunde. |

Rückgabewert

| Name | Typ | Beschreibung |
|--------------------|-----|--|
| DATE_and_TOD_To_DT | DT | Kombinierte Zeitinformation vom Typ DATE_AND_TIME. |

Beispiel

```
PROGRAM MAIN
VAR
    d          : DATE:= D#2022-09-15;
    t          : TOD:=TOD#17:24:56.789;
    dateAndTime : DT;
    wMilliseconds : WORD;
END_VAR

d:=D#2022-09-15;
t:=TOD#17:24:56.789;
dateAndTime:=DATE_and_TOD_To_DT(d:=d, t:=t, wMilliseconds=>wMilliseconds);
(* Result is: DT#2022-09-15-17:24:56, wMilliseconds = 789 *)

d:=D#2022-08-16;
t:=TOD#10:00:00.000;
dateAndTime:=DATE_and_TOD_To_DT(d:=d, t:=t, wMilliseconds=>wMilliseconds);
(* Result is: DT#2022-08-16-10:00:00, wMilliseconds = 0 *)
```

6.4.34 Date_To_BinaryTime6Day

Konvertiert ein Datum des Typs: DATE in die Anzahl der vergangenen Tage seit dem 1. Januar 1984 vom Typ WORD. Die Funktion liefert den Wert „0“ zurück, wenn das zu konvertierende Datum vor dem 1. Januar 1984 liegt.

Namensraum: Tc3_Collections [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
FUNCTION Date_To_BinaryTime6Day : WORD(0..16#AE35)
VAR_INPUT
    in      : DATE := D#1984-01-01;
END_VAR
VAR_OUTPUT
    bError  : BOOL;
END_VAR
```

Eingänge

| Name | Typ | Beschreibung |
|------|------|------------------------------|
| in | DATE | Das zu konvertierende Datum. |

Ausgänge

| Name | Typ | Beschreibung |
|--------|------|---|
| bError | BOOL | Der Ausgang ist „TRUE“ wenn der Eingangswert kleiner ist als „D#1984-01-01“. In diesem Fall liefert die Funktion den Rückgabewert „0“ zurück. |

Rückgabewert

| Name | Typ | Beschreibung |
|------------------------|------|--|
| Date_To_BinaryTime6Day | WORD | Konvertierte Anzahl der vergangenen Tage seit dem 1. Januar 1984. Dieser Wert entspricht dem Wert der T_BinaryTime [▶ 476].day-Komponente. |

Beispiel

```
METHOD FINAL BinaryTime6Day_ToFrom_Date : BOOL
VAR
    tValue : T_BinaryTime;
    d      : DATE;
    sValue : STRING;
END_VAR
```

```
tValue.timeOfDay:=TOD#00:00:00.125;
tValue.day:=Date_To_BinaryTime6Day(D#1984-01-01); (* tValue.day:=16#0000 *)
sValue:=BinaryTime6_To_String(tValue); (* sValue:='BT#1984-01-01-00:00:00.125' *)
d:=BinaryTime6Day_To_Date(tValue.day); (* d:=D#1984-01-01 *)
```

```
tValue.timeOfDay:=TOD#01:02:03.125;
tValue.day:=Date_To_BinaryTime6Day(D#1984-01-02); (* tValue.day:=16#0001 *)
sValue:=BinaryTime6_To_String(tValue); (* sValue:='BT#1984-01-02-01:02:03.125' *)
d:=BinaryTime6Day_To_Date(tValue.day); (* d:=D#1984-01-02 *)
```

```
tValue.timeOfDay:=TOD#04:05:06.125;
tValue.day:=Date_To_BinaryTime6Day(D#2018-03-21); (* tValue.day:=16#30D2 *)
sValue:=BinaryTime6_To_String(tValue); (* sValue:='BT#2018-03-21-04:05:06.125' *)
d:=BinaryTime6Day_To_Date(tValue.day); (* d:=D#2018-03-21 *)
```

```
tValue.timeOfDay:=TOD#07:08:09;
tValue.day:=tValue.day + 7; (* add 7 days *)
sValue:=BinaryTime6_To_String(tValue); (* sValue:='BT#2018-03-28-07:08:09' *)
```

```
d:=BinaryTime6Day_To_Date(tValue.day); (* d:=D#2018-03-28 *)
BinaryTime6Day_ToFrom_Date:=TRUE;
```

6.4.35 FillOctetString

Überschreibt alle Oktette eines Oktett-Strings mit einem vorgegebenen Wert.

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

```
FUNCTION FillOctetString : UDINT
VAR_IN_OUT
    o      : ARRAY[*] OF BYTE;
END_VAR
VAR_INPUT
    fillOctet : BYTE;
END_VAR
```

 /  **Ein-/Ausgänge**

| Name | Typ | Beschreibung |
|------|------------------|--|
| o | ARRAY[*] OF BYTE | Oktett-String der überschrieben werden soll. |

 **Eingänge**

| Name | Typ | Beschreibung |
|-----------|------|--|
| fillOctet | BYTE | Wert, mit dem die Oktette überschrieben werden sollen. |

 **Rückgabewert**

| Name | Typ | Beschreibung |
|-----------------|-------|-------------------------------------|
| FillOctetString | UDINT | Anzahl der überschriebenen Oktette. |

Beispiel

```
METHOD FINAL Sample_FillOctetString : BOOL
VAR
    o6   : T_OCTET6:= [6(0)];
    o8   : T_OCTET8:= [8(0)];
    o64  : T_OCTET64:= [64(0)];
    size : UDINT;
END_VAR
size:=FillOctetString(o:=o6, fillOctet:=16#FF); (* size:=6, all octets: 16#FF *)
size:=FillOctetString(o:=o8, fillOctet:=16#BB); (* size:=8, all octets: 16#BB *)
size:=FillOctetString(o:=o64, fillOctet:=16#CC); (* size:=64, all octets: 16#CC *)
Sample_FillOctetString:=TRUE;
```

6.4.36 INT24

Initialisiert eine vorzeichenbehaftete 24-Bit-Zahl des Typs: T_INT24.

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

```
FUNCTION INT24 : T_INT24
VAR_INPUT
    Hi : BYTE;
    Mi : BYTE;
    Lo : BYTE;
END_VAR
```

Eingänge

| Name | Typ | Beschreibung |
|------|------|--|
| Hi | BYTE | Obersten 8 Bits, die gesetzt werden sollen. |
| Mi | BYTE | Mittleren 8 Bits, die gesetzt werden sollen. |
| Lo | BYTE | Untersten 8 Bits, die gesetzt werden sollen. |

Rückgabewert

| Name | Typ | Beschreibung |
|-------|-----------------------------------|--------------|
| INT24 | T_INT24 [▶ 477] | 24-Bit-Zahl. |

Beispiel

```
METHOD FINAL Sample_INT24 : BOOL
VAR
    iValue: T_INT24;
END_VAR

iValue:=INT24(16#AB, 16#CD, 16#EF); (* iValue:=16#ABCDEF *)
Sample_INT24:=TRUE;
```

6.4.37 INT128

Initialisiert eine vorzeichenbehaftete 128-Bit-Zahl des Typs: T_INT128.

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

```
FUNCTION INT128 : T_INT128
VAR_INPUT
    Hi : LWORD;
    Lo : LWORD;
END_VAR
```

Eingänge

| Name | Typ | Beschreibung |
|------|-------|---|
| Hi | LWORD | Oberen 64 Bits, die gesetzt werden sollen. |
| Lo | LWORD | Unteren 64 Bits, die gesetzt werden sollen. |

Rückgabewert

| Name | Typ | Beschreibung |
|--------|------------------------------------|---------------|
| INT128 | T_INT128 [▶ 477] | 128-Bit-Zahl. |

Beispiel

```
METHOD FINAL Sample_INT128 : BOOL
VAR
    iValue: T_INT128;
END_VAR

iValue:=INT128(16#000000000000ABCDEF, 16#1234567890000001);
(* iValue:=16#000000000000ABCDEF_1234567890000001 *)
Sample_INT128:=TRUE;
```


6.4.38 LTIME_To_UtcTimeFractionOfSecond

Konvertiert den Bruchteil einer Sekunde des Typs LTIME in den Typ T_UINT24 [▶ 479]. Der konvertierte Wert entspricht dem Wert der T_UtcTime [▶ 479].fractionOfSecond-Komponente. Der LTIME-Wert wird in 24 Bits auf besondere Weise codiert. Siehe: Codierung der UTC-Time.fractionOfSecond0..23-Bits [▶ 682]. Der Eingangswert muss <= LTIME#999ms999us999ns. Die max. Auflösung bei der Konvertierung beträgt ~60ns.

Namensraum: Tc3_Collections [▶ 96]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
FUNCTION LTIME_To_UtcTimeFractionOfSecond : T_UINT24
VAR_INPUT
    in      : LTIME;
END_VAR
VAR_OUTPUT
    bOverflow    : BOOL;
    secondsOver  : LWORD;
    nanosec      : LWORD;
    attosec      : LWORD;
END_VAR
```

📡 Eingänge

| Name | Typ | Beschreibung |
|------|-------|--|
| in | LTIME | Der zu konvertierende Bruchteil einer Sekunde. |

📡 Ausgänge

| Name | Typ | Beschreibung |
|-------------|-------|--|
| bOverflow | BOOL | Zusätzlicher Ausgangsparameter der Information über den Eingangswertüberlauf liefert. Beim Eingangswert >= LTIME#1s ist dieser Wert „TRUE“, sonst „FALSE“. |
| secondsOver | LWORD | Zusätzlicher Ausgangsparameter der die Anzahl der Überlaufsekunden liefert. |
| nanosec | LWORD | Zusätzlicher Ausgangsparameter der den Eingangswert in Nanosekunden liefert. |
| attosec | LWORD | Zusätzlicher Ausgangsparameter der den Eingangswert in Attosekunden liefert. |

📡 Rückgabewert

| Name | Typ | Beschreibung |
|----------------------------------|---------------------|---|
| LTIME_To_UtcTimeFractionOfSecond | T_UINT24
[▶ 479] | Konvertierter Bruchteil einer Sekunde als Binärdaten. |

Beispiel

```
METHOD UtcTimeFractionOfSecond_ToFrom_LTIME : BOOL
VAR
    tValue    : T_UtcTime;
    sValue    : STRING;
    tFos      : LTIME;
    nanosec   : LWORD;
END_VAR

tValue.quality := String_To_UtcTimeQuality('UQ#000|24');
tValue.secondsSinceEpoch := DT#2018-03-28-12:13:14;

tValue.fractionOfSecond := LTIME_To_UtcTimeFractionOfSecond(LTIME#0NS);
(* tValue.fractionOfSecond := [16#00, 16#00, 16#00], 0 significant bits *)
sValue := UtcTime_To_String(tValue); (* sValue := 'UT#2018-03-28-12:13:14.000000000|000|24' *)
```

```

tFos := UtcTimeFractionOfSecond_To_LTIME(tValue.fractionOfSecond, nanosec=>nanosec);
(* tFos := LTIME#0ns, nanosec := 0 *)

tValue.fractionOfSecond := LTIME_To_UtcTimeFractionOfSecond(LTIME#750MS);
(* tValue.fractionOfSecond := [16#03, 16#00, 16#00], 3 significant bits *)
sValue := UtcTime_To_String(tValue);(* sValue := 'UT#2018-03-28-12:13:14.750000000|000|24' *)
tFos := UtcTimeFractionOfSecond_To_LTIME(tValue.fractionOfSecond, nanosec=>nanosec);
(* tFos := LTIME#750ms, nanosec := 750000000 *)

tValue.fractionOfSecond := LTIME_To_UtcTimeFractionOfSecond(LTIME#968MS750US);
(* tValue.fractionOfSecond := [16#1F, 16#00, 16#00], 5 significant bits *)
sValue := UtcTime_To_String(tValue);(* sValue := 'UT#2018-03-28-12:13:14.968750000|000|24' *)
tFos := UtcTimeFractionOfSecond_To_LTIME(tValue.fractionOfSecond, nanosec=>nanosec);
(* tFos := LTIME#968ms750us, nanosec := 968750000 *)

tValue.fractionOfSecond := LTIME_To_UtcTimeFractionOfSecond(LTIME#998MS46US875NS);
(* tValue.fractionOfSecond := [16#FF, 16#01, 16#00], 9 significant bits *)
sValue := UtcTime_To_String(tValue);(* sValue := 'UT#2018-03-28-12:13:14.998046875|000|24' *)
tFos := UtcTimeFractionOfSecond_To_LTIME(tValue.fractionOfSecond, nanosec=>nanosec);
(* tFos := LTIME#998ms46us875ns, nanosec := 998046875 *)

tValue.fractionOfSecond := LTIME_To_UtcTimeFractionOfSecond(LTIME#999MS999US999NS);
(* tValue.fractionOfSecond := [16#FF, 16#FF, 16#FF], 24 significant bits *)
sValue := UtcTime_To_String(tValue);(* sValue := 'UT#2018-03-28-12:13:14.999999940|000|24' *)
tFos := UtcTimeFractionOfSecond_To_LTIME(tValue.fractionOfSecond, nanosec=>nanosec);
(* tFos := LTIME#999ms999us940ns, nanosec := 999999940 *)

UtcTimeFractionOfSecond_ToFrom_LTIME := TRUE;

```

6.4.39 LWORD_To_BinaryTime6

Konvertiert die Binärdaten des Typs LWORD in die Komponenten des strukturierten Typs [T_BinaryTime](#) [► 476]. Die Kodierung der Komponenten im LWORD entspricht der TwinCAT [Binary-Time-LWORD-Formatspezifikation](#) [► 681].

Namensraum: [Tc3_Collections](#) [► 96]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```

FUNCTION LWORD_To_BinaryTime6 : T_BinaryTime
VAR_INPUT
    in      : LWORD(0..16#FFFFFFFFFFFFFF);
END_VAR
VAR_OUTPUT
    bError : BOOL;
END_VAR

```

Eingänge

| Name | Typ | Beschreibung |
|------|-----------------------------|---|
| in | LWORD(0..16#FFFFFFFFFFFFFF) | Die zu konvertierende Zeitinformation als Binary-Time-Binärdaten. |

Ausgänge

| Name | Typ | Beschreibung |
|--------|------|--|
| bError | BOOL | Der Ausgang ist „TRUE“ wenn der Wert der Bits 0..31 (timeOfDay-Komponente) größer ist als „TOD#23:59:59.999“. In diesem Fall liefert die Funktion den Rückgabewert „STRUCT(timeOfDay := TOD#00:00:00, day := 0)“ zurück. |

Rückgabewert

| Name | Typ | Beschreibung |
|----------------------|--------------------------------------|--|
| LWORD_To_BinaryTime6 | T_BinaryTime [► 476] | Konvertierte Zeitinformation als Binary-Time-Struktur. |

Beispiel

```
METHOD BinaryTime6_ToFrom_LWORD : BOOL
VAR_INPUT
    fbIED : REFERENCE TO FB_IED;
END_VAR
VAR
    tValue : T_BinaryTime;
    sValue : STRING;
END_VAR

IF NOT __ISVALIDREF(fbIED) THEN
    RETURN;
END_IF

tValue := LWORD_To_BinaryTime6(0);
sValue := BinaryTime6_To_String(tValue); (* sValue := 'BT#1984-01-01-00:00:00' *)
IF fbIED.Relay.LLN0.brcb1.TimeOfEntry.nValue <> BinaryTime6_To_LWORD(tValue) THEN
    tValue := fbIED.Relay.LLN0.brcb1.TimeOfEntry.tValue;
END_IF

tValue := LWORD_To_BinaryTime6(1);
sValue := BinaryTime6_To_String(tValue); (* sValue := 'BT#1984-01-01-00:00:00.001' *)
IF fbIED.Relay.LLN0.brcb2.TimeOfEntry.nValue <> BinaryTime6_To_LWORD(tValue) THEN
    tValue := fbIED.Relay.LLN0.brcb2.TimeOfEntry.tValue;
END_IF

tValue := String_To_BinaryTime6('BT#2017-03-21-11:22:33.44');
sValue := BinaryTime6_To_String(tValue); (* sValue := 'BT#2017-03-21-11:22:33.440' *)
IF fbIED.Relay.LLN0.brcb3.TimeOfEntry.nValue <> BinaryTime6_To_LWORD(tValue) THEN
    tValue := fbIED.Relay.LLN0.brcb3.TimeOfEntry.tValue;
END_IF

BinaryTime6_ToFrom_LWORD := TRUE;
```

6.4.40 LWORD_TO_MACADDR

Konvertiert die Binärdaten des Typs LWORD in den Typ ETHERNET_ADDRESS. Die LWORD-Bytes werden bei der Konvertierung nach dem folgenden Schema konvertiert und getauscht: Das niederwertigste Byte wird zum letzten und das höherwertigste, konvertierte Byte wird zum ersten Oktett der MAC-Adresse.

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

```
FUNCTION LWORD_TO_MACADDR : ETHERNET_ADDRESS
VAR_INPUT
    in: LWORD(0..16#FFFFFFFFFFFFFF);
END_VAR
```

 **Eingänge**

| Name | Typ | Beschreibung |
|------|-----------------------------|-----------------------------------|
| in | LWORD(0..16#FFFFFFFFFFFFFF) | Der zu konvertierende LWORD Wert. |

 **Rückgabewert**

| Name | Typ | Beschreibung |
|------------------|--|---|
| LWORD_TO_MACADDR | ETHERNET_ADDRESS [▶ 451] | Oktett-String mit der max. Länge von 8 Oktetts. |

Beispiel

```
METHOD FINAL Sample_LWORD_TO_MACADDR : BOOL
VAR
    mac: ETHERNET_ADDRESS;
END_VAR

mac:=LWORD_TO_MACADDR(16#00B0D063C226);
(* mac:=00-B0-D0-63-C2-26 *)
Sample_LWORD_TO_MACADDR:=TRUE;
```

6.4.41 LWORD_TO_OCTET6

Konvertiert die Binärdaten des Typs LWORD in den Typ T_OCTET6. Die zwei höchstwertigen Bytes werden ignoriert und nicht konvertiert. Die übrigen LWORD-Bytes werden bei der Konvertierung nach dem folgenden Schema konvertiert und getauscht: Das niederwertigste Byte wird zum letzten und das höherwertigste, konvertierte Byte wird zum ersten Oktett des Oktett-Strings.

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

```
FUNCTION LWORD_TO_OCTET6 : T_OCTET6
VAR_INPUT
    in: LWORD(0..16#FFFFFFFFFFFF);
END_VAR
```

Eingänge

| Name | Typ | Beschreibung |
|------|---------------------------|-----------------------------------|
| in | LWORD(0..16#FFFFFFFFFFFF) | Der zu konvertierende LWORD Wert. |

Rückgabewert

| Name | Typ | Beschreibung |
|-----------------|------------------------------------|---|
| LWORD_TO_OCTET6 | T_OCTET6 [▶ 478] | Oktett-String mit der max. Länge von 6 Oktetts. |

Beispiel

```
METHOD FINAL Sample_LWORD_TO_OCTET6 : BOOL
VAR
    o: T_OCTET6;
END_VAR

o:=LWORD_TO_OCTET6(16#010CCD010001);
(* o:=[16#01,16#0C,16#CD,16#01,16#00,16#01] *)
Sample_LWORD_TO_OCTET6:=TRUE;
```

6.4.42 LWORD_TO_OCTET8

Konvertiert die Binärdaten des Typs LWORD in den Typ T_OCTET8. Die LWORD-Bytes werden bei der Konvertierung nach dem folgenden Schema konvertiert und getauscht: Das niederwertigste Byte wird zum letzten und das höherwertigste, konvertierte Byte wird zum ersten Oktett des Oktett-Strings.

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

```
FUNCTION LWORD_TO_OCTET8 : T_OCTET8
VAR_INPUT
    in: LWORD;
END_VAR
```

Eingänge

| Name | Typ | Beschreibung |
|------|-------|-----------------------------------|
| in | LWORD | Der zu konvertierende LWORD Wert. |

Rückgabewert

| Name | Typ | Beschreibung |
|-----------------|------------------------------------|---|
| LWORD_TO_OCTET8 | T_OCTET8 [▶ 478] | Oktett-String mit der max. Länge von 8 Oktetts. |

Beispiel

```
METHOD FINAL Sample_LWORD_TO_OCTET8 : BOOL
VAR
  o: T_OCTET8;
END_VAR

o:=LWORD_TO_OCTET8(16#0102030405060708);
(* o:=[16#01,16#02,16#03,16#04,16#05,16#06,16#07,16#08] *)
Sample_LWORD_TO_OCTET8:=TRUE;
```

6.4.43 LWORD_To_UtcTime

Konvertiert die Binärdaten des Typs LWORD in die Komponenten des strukturierten Typs [T_UtcTime](#) [[▶ 479](#)]. Die Kodierung der Komponenten im LWORD entspricht der TwinCAT [UTC-Time-LWORD-Formatspezifikation](#) [[▶ 682](#)].

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]
Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
FUNCTION LWORD_To_UtcTime : T_UtcTime
VAR_INPUT
  in : LWORD;
END_VAR
```

 **Eingänge**

| Name | Typ | Beschreibung |
|------|-------|--|
| in | LWORD | Die zu konvertierende Zeitinformation als UTC-Time-Binärdaten. |

 **Rückgabewert**

| Name | Typ | Beschreibung |
|------------------|---|---|
| LWORD_To_UtcTime | T_UtcTime [▶ 479] | Konvertierte Zeitinformation als UTC-Time-Struktur. |

Beispiel

```
METHOD UtcTime_ToFrom_LWORD : BOOL
VAR
  tValue : T_UtcTime;
  sValue : STRING;
END_VAR

tValue := LWORD_To_UtcTime(0);
sValue := UtcTime_To_String(tValue);(* sValue := 'UT#1970-01-01-00:00:00.000000000|000|0' *)

IF UtcTime_To_LWORD(tValue) = 0 THEN
  tValue.secondSinceEpoch := DT#2000-01-01-00:00:0;
  tValue.fractionOfSecond := LTIME_TO_UTCtimeFractionOfSecond( LTIME#0MS );
  tValue.quality.LeapSecondsKnown := FALSE;
  tValue.quality.ClockFailure := FALSE;
  tValue.quality.ClockNotSynchronized := TRUE;
  Accuracy_To_UTCtimeQualityAccuracy( E_UTCtimeAccuracy_03,
    bAccuracy0=>tValue.quality.Accuracy0,
    bAccuracy1=>tValue.quality.Accuracy1,
    bAccuracy2=>tValue.quality.Accuracy2,
    bAccuracy3=>tValue.quality.Accuracy3,
    bAccuracy4=>tValue.quality.Accuracy4 );
  sValue := UtcTime_To_String(tValue);(* sValue := 'UT#2000-01-01-00:00:00.000000000|001|3' *)
END_IF

UtcTime_ToFrom_LWORD := TRUE;
```

6.4.44 OctetString_To_AnyBuffer

Konvertiert und kopiert die Daten von einem Oktett-String oder einem eindimensionalen Byte-Array in eine beliebige Variable.

Namensraum: Tc3_Collections [▶ 96]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

```
FUNCTION OctetString_To_AnyBuffer : UDINT
VAR_IN_OUT
    source      : ARRAY[*] OF BYTE;
END_VAR
VAR_INPUT
    pTarget     : PVOID;
    cbTarget    : UDINT;
    bSwapBytes  : BOOL;
END_VAR
VAR_OUTPUT
    bUnderflow  : BOOL;
    bOverflow   : BOOL;
END_VAR
```

Ein-/Ausgänge

| Name | Typ | Beschreibung |
|--------|------------------|---------------|
| source | ARRAY[*] OF BYTE | Quellvariable |

Eingänge

| Name | Typ | Beschreibung |
|------------|-------|---|
| pTarget | PVOID | Adresszeiger der Zielvariablen. |
| cbTarget | UDINT | Bytegröße der Zielvariablen. |
| bSwapBytes | BOOL | Bei TRUE werden die Datenbytes getauscht. |

Ausgänge

| Name | Typ | Beschreibung |
|------------|------|--|
| bUnderflow | BOOL | Wenn TRUE dann ist die Quellvariable kleiner als die Zielvariable. |
| bOverflow | BOOL | Wenn TRUE dann ist die Quellvariable größer als die Zielvariable. |

Rückgabewert

| Name | Typ | Beschreibung |
|--------------------------|-------|--|
| OctetString_To_AnyBuffer | UDINT | Anzahl der erfolgreich kopierten Datenbytes. |

Beispiel

```
METHOD FINAL Sample_OctetString_To_AnyBuffer : BOOL
VAR
    sValue      : STRING;
    ui64        : LWORD;
    ui32        : UDINT;
    o6          : T_OCTET6:=[16#30, 16#31, 16#32, 16#33, 16#34, 16#35];
    o8          : T_OCTET8:=[16#30, 16#31, 16#32, 16#33, 16#34, 16#35, 16#36, 16#37];
    o64        : T_OCTET64:=[16#30, 16#31, 16#32, 16#33, 16#34, 16#35, 16#36, 16#37, 16#38, 16#39];
    size        : UDINT;
    bOverflow   : BOOL;
    bUnderflow  : BOOL;
END_VAR
size:=OctetString_To_AnyBuffer(source:=o6, pTarget:=ADR(sValue), cbTarget:=SIZEOF(sValue), bSwapByte
s:=FALSE, bUnderflow=>bUnderflow, bOverflow=>bOverflow);(* size:=6 *)
size:=OctetString_To_AnyBuffer(source:=o8, pTarget:=ADR(ui64), cbTarget:=SIZEOF(ui64), bSwapBytes:=T
RUE, bUnderflow=>bUnderflow, bOverflow=>bOverflow);(* size:=8 *)
size:=OctetString_To_AnyBuffer(source:=o64, pTarget:=ADR(ui32), cbTarget:=SIZEOF(ui32), bSwapBytes:=
```

```
TRUE, bUnderflow=>bUnderflow, bOverflow=>bOverflow); (* size:=4 *)
Sample_OctetString_To_AnyBuffer:=(sValue = '012345') AND (ui64 = 16#3031323334353637) AND (ui32 = 16#30313233);
```

6.4.45 SizeOfOctetString

Ermittelt die Bytegröße eines beliebigen Oktett-Strings (1-Dimensionales Byte-Array).

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

```
FUNCTION SizeOfOctetString : UDINT
VAR_IN_OUT
  o : ARRAY[*] OF BYTE;
END_VAR
```

 **Ein-/Ausgänge**

| Name | Typ | Beschreibung |
|------|------------------|---|
| o | ARRAY[*] OF BYTE | Oktett-String dessen Größe ermittelt werden soll. |

 **Rückgabewert**

| Name | Typ | Beschreibung |
|-------------------|-------|--------------------------|
| SizeOfOctetString | UDINT | Oktett-String Bytegröße. |

Beispiel

```
METHOD FINAL Sample_SizeOfOctetString : BOOL
VAR
  o6 : T_OCTET6:=[6(0)];
  o8 : T_OCTET8:=[8(0)];
  o64 : T_OCTET64:=[64(0)];
  size: UDINT;
END_VAR

size:=SizeOfOctetString(o:=o6); (* size:=6 *)
size:=SizeOfOctetString(o:=o8); (* size:=8 *)
size:=SizeOfOctetString(o:=o64); (* size:=64 *)
Sample_SizeOfOctetString:=TRUE;
```

6.4.46 Source_To_AcsiQualitySource

Konvertiert den Wert des Aufzählungstyps [E_AcsiQualitySource](#) [[▶ 424](#)] in einen boolischen Wert. Dieser Ausgangswert entspricht der Source-Bit-Komponente des strukturierten Typs [ST_AcsiQuality](#) [[▶ 463](#)].

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
FUNCTION Source_To_AcsiQualitySource : BOOL
VAR_INPUT
  in : E_AcsiQualitySource;
END_VAR
```

 **Eingänge**

| Name | Typ | Beschreibung |
|------|---|---|
| in | E_AcsiQualitySource [▶ 424] | Die zu konvertierende Information als Aufzählungstyp. |

Rückgabewert

| Name | Typ | Beschreibung |
|-----------------------------|------|---|
| Source_To_AcsiQualitySource | BOOL | Konvertierte Information als ST_AcsiQuality [▶ 463]. Source-Bit-Komponente. |

Beispiel

```

METHOD AcsiQualitySource_ToFrom_Source : BOOL
VAR_INPUT
    fbIED : REFERENCE TO FB_IED;
END_VAR
VAR
    qValue : ST_AcsiQuality;
    sValue : STRING;
END_VAR

IF NOT __ISVALIDREF(fbIED) THEN
    RETURN;
END_IF

Validity_To_AcsiQualityValidity(E_AcsiQualityValidity.Good, bValidity0=>qValue.Validity0, bValidity1
=> qValue.Validity1 );
qValue.Overflow := FALSE;
qValue.OutOfRange := FALSE;
qValue.BadReference := FALSE;
qValue.Oscillatory := FALSE;
qValue.Failure := FALSE;
qValue.OldData := FALSE;
qValue.Inconsistent := FALSE;
qValue.Inaccurate := FALSE;
qValue.Source := Source_To_AcsiQualitySource(E_AcsiQualitySource.Substituted);
qValue.Test := TRUE;
qValue.OperatorBlocked := FALSE;
sValue := AcsiQuality_To_String(qValue);(* sValue := 'Q#00|00000000|1|10' *)

IF fbIED.Relay.LLN0.Beh.q.eSource <> AcsiQualitySource_To_Source(qValue) THEN
    qValue := fbIED.Relay.LLN0.Beh.q.qValue;
END_IF

AcsiQualitySource_ToFrom_Source := TRUE;

```

6.4.47 String_To_AcsiQuality

Konvertiert die als String formatierte Quality-Information in die Komponenten des strukturierten Typs [ST_AcsiQuality](#) [▶ 463]. Die Zeichenfolge des zu konvertierenden Strings ist vorgegeben und hat folgenden Aufbau: 'Q#vv|ORBSFDCA|s|TB'. Sie entspricht der TwinCAT [Quality-String-Formatspezifikation](#) [▶ 687].

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

FUNCTION String_To_AcsiQuality : ST_AcsiQuality
VAR_INPUT
    in : STRING(18) := 'Q#00|00000000|0|00';
END_VAR
VAR_OUTPUT
    bError : BOOL;
END_VAR

```

Eingänge

| Name | Typ | Beschreibung |
|------|------------|---|
| in | STRING(18) | Die zu konvertierende Information als Quality-String. |

 **Ausgänge**

| Name | Typ | Beschreibung |
|--------|------|---|
| bError | BOOL | Dieser Ausgang ist TRUE wenn die Zeichenfolge des Eingangsstrings nicht fehlerfrei konvertiert werden konnte. |

 **Rückgabewert**

| Name | Typ | Beschreibung |
|-----------------------|--------------------------------------|--|
| String_To_AcsiQuality | ST_AcsiQuality ▶ 463 | Konvertierte Information als Quality-Struktur. |

Beispiel

```
METHOD AcsiQuality_ToFrom_String : BOOL
VAR_INPUT
    fbIED : REFERENCE TO FB_IED;
END_VAR
VAR
    qValue : ST_AcsiQuality;
    sValue : STRING;
END_VAR

IF NOT __ISVALIDREF(fbIED) THEN
    RETURN;
END_IF

qValue := String_To_AcsiQuality('Q#00|00000000|0|00');
sValue := AcsiQuality_To_String(qValue); (* sValue := 'Q#00|00000000|0|00' *)
IF fbIED.Relay.LLN0.Beh.q.sValue <> AcsiQuality_To_String(qValue) THEN
    qValue := fbIED.Relay.LLN0.Beh.q.qValue;
END_IF

qValue := String_To_AcsiQuality('Q#01|00110011|0|01');
sValue := AcsiQuality_To_String(qValue); (* sValue := 'Q#01|00110011|0|01' *)
IF fbIED.Relay.LLN0.Health.q.sValue <> AcsiQuality_To_String(qValue) THEN
    qValue := fbIED.Relay.LLN0.Health.q.qValue;
END_IF

qValue := String_To_AcsiQuality('Q#10|11001100|1|10');
sValue := AcsiQuality_To_String(qValue); (* sValue := 'Q#10|11001100|1|10' *)
IF fbIED.Relay.LLN0.Mod_.q.sValue <> AcsiQuality_To_String(qValue) THEN
    qValue := fbIED.Relay.LLN0.Mod_.q.qValue;
END_IF

AcsiQuality_ToFrom_String := TRUE;
```

6.4.48 String_To_BinaryTime6

Konvertiert die als String formatierte Binary-Time-Zeitinformation in die Komponenten des strukturierten Typs [T_BinaryTime](#) [▶ 476](#). Die Zeichenfolge des zu konvertierenden Strings ist vorgegeben und hat folgenden Aufbau: 'BT#[YYYY-MM-DD-]hh:mm:ss[.n[n[n]]]'. Sie entspricht der TwinCAT [Binary-Time-String-Formatspezifikation](#) [▶ 681](#).

Namensraum: [Tc3_Collections](#) [▶ 96](#)

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
FUNCTION String_To_BinaryTime6 : T_BinaryTime
VAR_INPUT
    in : STRING(26);
END_VAR
VAR_OUTPUT
    bError : BOOL;
END_VAR
```

Eingänge

| Name | Typ | Beschreibung |
|------|------------|---|
| in | STRING(26) | Die zu konvertierende Zeitinformation als Binary-Time-String. |

Ausgänge

| Name | Typ | Beschreibung |
|--------|------|--|
| bError | BOOL | Dieser Ausgang ist „TRUE“ wenn die Zeichenfolge des Eingangsstrings nicht fehlerfrei konvertiert werden konnte. In diesem Fall liefert die Funktion den Rückgabewert „STRUCT(timeOfDay := TOD#00:00:00, day := 0)“ zurück. |

Rückgabewert

| Name | Typ | Beschreibung |
|-----------------------|--|--|
| String_To_BinaryTime6 | T_BinaryTime 476 | Konvertierte Zeitinformation als Binary-Time-Struktur. |

Beispiel

```

METHOD BinaryTime6_ToFrom_String : BOOL
VAR_INPUT
    fbIED : REFERENCE TO FB_IED;
END_VAR
VAR
    tValue : T_BinaryTime;
    sValue : STRING;
END_VAR

IF NOT __ISVALIDREF(fbIED) THEN
    RETURN;
END_IF

tValue := String_To_BinaryTime6('BT#01:02:03');
sValue := BinaryTime6_To_String(tValue); (* sValue := 'BT#1984-01-01-01:02:03' *)
IF fbIED.Relay.LLN0.brcb1.TimeOfEntry.sValue <> sValue THEN
    tValue := fbIED.Relay.LLN0.brcb1.TimeOfEntry.tValue;
END_IF

tValue := String_To_BinaryTime6('BT#04:05:06.7');
sValue := BinaryTime6_To_String(tValue); (* sValue := 'BT#1984-01-01-04:05:06.700' *)
IF fbIED.Relay.LLN0.brcb2.TimeOfEntry.sValue <> sValue THEN
    tValue := fbIED.Relay.LLN0.brcb2.TimeOfEntry.tValue;
END_IF

tValue := String_To_BinaryTime6('BT#2017-01-18-11:12:13');
sValue := BinaryTime6_To_String(tValue); (* sValue := 'BT#2017-01-18-11:12:13' *)
IF fbIED.Relay.LLN0.brcb3.TimeOfEntry.sValue <> sValue THEN
    tValue := fbIED.Relay.LLN0.brcb3.TimeOfEntry.tValue;
END_IF

BinaryTime6_ToFrom_String := TRUE;

```

6.4.49 String_To_UtcTime

Konvertiert die als String formatierte UTC-Time-Zeitinformation in die Komponenten des strukturierten Typs [T_UtcTime](#) | [479](#). Die Zeichenfolge des zu konvertierenden Strings ist vorgegeben und hat folgenden Aufbau: 'UT#YYYY-MM-DD-hh:mm:ss.nnnnnnnn|LFC|A[A]'. Sie entspricht der TwinCAT [UTC-Time-String-Formatspezifikation](#) | [686](#).

Namensraum: [Tc3_Collections](#) | [96](#)

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
FUNCTION String_To_UtcTime : T_UTCtime
VAR_INPUT
    in      : STRING(39) := 'UT#1970-01-01-00:00:00.000000000|000|0';
END_VAR
VAR_OUTPUT
    bError  : BOOL;
END_VAR
```

 **Eingänge**

| Name | Typ | Beschreibung |
|------|------------|--|
| in | STRING(39) | Die zu konvertierende Zeitinformation als UTC-Time-String. |

 **Ausgänge**

| Name | Typ | Beschreibung |
|--------|------|---|
| bError | BOOL | Dieser Ausgang ist „TRUE“ wenn die Zeichenfolge des Eingangsstrings nicht fehlerfrei konvertiert werden konnte. |

 **Rückgabewert**

| Name | Typ | Beschreibung |
|-------------------|---------------------------------|---|
| String_To_UTCtime | T_UTCtime ▶ 479 | Konvertierte Zeitinformation als UTC-Time-Struktur. |

Beispiel

'UT#2017-01-16-11:22:33.750000000|001|24' bedeutet Datum: „2017-01-16“, Uhrzeit: „11:22:33“ und „750“ ms, Schaltsekunden unbekannt, Zeitgeber hat keinen Fehler, Uhrzeit ist nicht synchronisiert.

```
METHOD UtcTime_ToFrom_String : BOOL
VAR_INPUT
    fbIED : REFERENCE TO FB_IED;
END_VAR
VAR
    tValue : T_UTCtime;
    sValue : STRING;
END_VAR

IF NOT __ISVALIDREF(fbIED) THEN
    RETURN;
END_IF

tValue := String_To_UTCtime('UT#1970-01-01-00:00:00.000000000|000|0');
sValue := UtcTime_To_String(tValue); (* sValue := 'UT#1970-01-01-00:00:00.000000000|000|0' *)
IF fbIED.Relay.LLN0.Beh.t.sValue <> sValue THEN
    tValue := fbIED.Relay.LLN0.Beh.t.tValue;
END_IF

tValue := String_To_UTCtime('UT#2018-03-21-14:56:55.125000000|010|3');
(* ClockFailure := 1, accuracy := 3 bits *)
sValue := UtcTime_To_String(tValue); (* sValue := 'UT#2018-03-21-14:56:55.125000000|010|3' *)
IF fbIED.Relay.LLN0.Health.t.sValue <> sValue THEN
    tValue := fbIED.Relay.LLN0.Health.t.tValue;
END_IF

tValue := String_To_UTCtime('UT#2017-01-16-11:22:33.750000000|001|24');
(* ClockNotSynchronized := 1, accuracy := 24 bits *)
sValue := UtcTime_To_String(tValue); (* sValue := 'UT#2017-01-16-11:22:33.750000000|001|24' *)
IF fbIED.Relay.LLN0.Mod_.t.sValue <> sValue THEN
    tValue := fbIED.Relay.LLN0.Mod_.t.tValue;
END_IF

UtcTime_ToFrom_String := TRUE;
```

6.4.50 String_To_UtcTimeQuality

Konvertiert die als String formatierte UTC-Time-Quality-Information in die Komponenten des strukturierten Typs `T_UtcTimeQuality` [▶ 480]. Die Zeichenfolge des zu konvertierenden Strings ist vorgegeben und hat folgenden Aufbau: 'UQ#LFC|A[A]'. Sie entspricht der TwinCAT UTC-Time-Quality-String-Formatspezifikation [▶ 685].

Namensraum: `Tc3_Collections` [▶ 96]

Bibliothek: `Tc3_Collections` (`Tc3_Collections.compiled-library`)

Syntax

```
FUNCTION String_To_UtcTimeQuality : T_UtcTimeQuality
VAR_INPUT
    in      : STRING(9) := 'UQ#000|0';
END_VAR
VAR_OUTPUT
    bError  : BOOL;
END_VAR
```

👉 Eingänge

| Name | Typ | Beschreibung |
|------|-----------|--|
| in | STRING(9) | Die zu konvertierende Information als UTC-Time-Quality-String. |

👉 Ausgänge

| Name | Typ | Beschreibung |
|--------|------|--|
| bError | BOOL | Dieser Ausgang ist „TRUE“ wenn die Zeichenfolge des Eingangsstring nicht fehlerfrei konvertiert werden konnte. |

👉 Rückgabewert

| Name | Typ | Beschreibung |
|--------------------------|--|---|
| String_To_UtcTimeQuality | <code>T_UtcTimeQuality</code>
[▶ 480] | Konvertierte Information als UTC-Time-Quality-Struktur. |

Beispiel

Der Rückgabewert: 'UQ#001|3' bedeutet, dass Schaltsekunden nicht bekannt sind. Zeitgeber hat keinen Fehler und ist nicht synchronisiert, die Auflösung der Zeitinformation des Bruchteils der Sekunde beträgt 3 Bits.

```
METHOD UtcTimeQuality_ToFrom_String : BOOL
VAR_INPUT
    fbIED      : REFERENCE TO FB_IED;
END_VAR
VAR
    Quality    : T_UtcTimeQuality;
    sQuality   : STRING;
END_VAR

IF NOT __ISVALIDREF(fbIED) THEN
    RETURN;
END_IF

Quality := String_To_UtcTimeQuality('UQ#000|0');
sQuality := UtcTimeQuality_To_String(Quality); (* sQuality := 'UQ#000|0' *)
IF fbIED.Relay.LLN0.Beh.t.sQuality <> sQuality THEN
    Quality := fbIED.Relay.LLN0.Beh.t.Quality;
END_IF

Quality := String_To_UtcTimeQuality('UQ#001|3');
sQuality := UtcTimeQuality_To_String(Quality); (* sQuality := 'UQ#001|3' *)
IF fbIED.Relay.LLN0.Health.t.sQuality <> sQuality THEN
```

```

Quality := fbIED.Relay.LLN0.Health.t.Quality;
END_IF

Quality := String_To_UtcTimeQuality('UQ#000|31');
sQuality := UtcTimeQuality_To_String(Quality); (* sQuality := 'UQ#000|31' *)
IF fbIED.Relay.LLN0.Mod_.t.sQuality <> sQuality THEN
    Quality := fbIED.Relay.LLN0.Mod_.t.Quality;
END_IF

UtcTimeQuality_ToFrom_String := TRUE;
    
```

6.4.51 String_To_UtcTimeStamp

Konvertiert die als String formatierte UTC-Zeitstempelinformation in die Komponenten des strukturierten Typs: [T_UtcTime](#) [▶ 479]. Die Zeichenfolge des zu konvertierenden Strings ist vorgegeben und hat folgenden Aufbau: 'US#YYYY-MM-DD-hh:mm:ss.nnnnnnnnn' wo:

- YYYY:=Jahr
- MM:=Monat
- DD:=Tag
- hh:=Stunde
- mm:=Minuten
- ss:=Sekunden

nnnnnnnnn:=Nanosekunden (9 Dezimalstellen). Die [T_UtcTime](#) [▶ 479].Quality-Komponente wird bei der Konvertierung nicht gesetzt (alle Bits haben den Wert 0).

Namensraum: [Tc3_Collections](#) [▶ 96]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```

FUNCTION String_To_UtcTimeStamp : T_UtcTime
VAR_INPUT
    in          : STRING(32) := 'US#1970-01-01-00:00:00.000000000';
END_VAR
VAR_OUTPUT
    bError      : BOOL;
    secondSinceEpoch : DT;
    fractionOfSecond : T_UINT24;
END_VAR
    
```

Eingänge

| Name | Typ | Beschreibung |
|------|------------|--|
| in | STRING(32) | Die zu konvertierende Zeitinformation als formatierter UTC-Zeitstempel-String. |

Ausgänge

| Name | Typ | Beschreibung |
|------------------|----------------------------------|--|
| bError | BOOL | Der Ausgang ist „TRUE“ wenn der Eingangswert nicht fehlerfrei konvertiert werden konnte. |
| secondSinceEpoch | DT | Wert der konvertierten T_UtcTime [▶ 479].secondSinceEpoch Komponente. |
| fractionOfSecond | T_UINT24 [▶ 479] | Wert der konvertierten T_UtcTime [▶ 479].fractionOfSecond Komponente. |

Rückgabewert

| Name | Typ | Beschreibung |
|------------------------|-----------------------------------|--|
| String_To_UtcTimeStamp | T_UtcTime [▶ 479] | Konvertierte Zeitstempelinformation als UTC-Time-Struktur. |

Beispiel

```

PROGRAM MAIN
VAR
  tUtc          : T_UtcTime;
  sUtc          : STRING;
  bError       : BOOL;
  secondsSinceEpoch : DT;
  fractionOfSecond : T_UINT24;
END_VAR

tUtc:=String_To_UtcTimeStamp(in:='US#2022-09-15-16:12:34.750000000' , bError=>bError, secondsSinceEpoch=>secondsSinceEpoch, fractionOfSecond=>fractionOfSecond);
sUtc:=UtcTimeStamp_To_String(in:=tUtc, bError=>bError);
(* => sUtc = 'US#2022-09-15-16:12:34.750000000' *)

tUtc.secondsSinceEpoch:=DT#2022-09-15-00:00:00;
tUtc.fractionOfSecond[0]:=0;
tUtc.fractionOfSecond[1]:=0;
tUtc.fractionOfSecond[2]:=0;
sUtc:=UtcTimeStamp_To_String(in:=tUtc, bError=>bError);
(* => sUtc = 'US#2022-09-15-00:00:00.000000000' *)

```

6.4.52 SystemTime_To_BinaryTime

Konvertiert die Zeitinformation des Typs [TIMESTRUCT](#) [[454](#)] (System-Time-Format) in die Zeitinformation des Typs [T_BinaryTime](#) [[476](#)] (Binary-Time-Format). Beim Konvertierungsfehler hat die Ausgangsvariable „bOverflow“ den Wert „TRUE“ und der zurückgelieferte Binary-Time-Wert das Datum:1984-01-01 und die Uhrzeit: 0h0m0s0ms.

Namensraum: [Tc3_Collections](#) [[96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```

FUNCTION SystemTime_To_BinaryTime : T_BinaryTime
VAR_INPUT
  in          : TIMESTRUCT;
END_VAR
VAR_OUTPUT
  bOverflow   : BOOL;
END_VAR

```

 Eingänge

| Name | Typ | Beschreibung |
|------|--|---|
| in | TIMESTRUCT [454] | Die zu konvertierende Zeitinformation als System-Time-Struktur. |

 Ausgänge

| Name | Typ | Beschreibung |
|-----------|------|---|
| bOverflow | BOOL | Datumsüberlauf wenn "TRUE". In diesem Fall hat die zu konvertierende System-Time ein Datum welches größer ist als 2106-02-07. |

 Rückgabewert

| Name | Typ | Beschreibung |
|--------------------------|--|--|
| SystemTime_To_BinaryTime | T_BinaryTime [476] | Konvertierte Zeitinformation als Binary-Time-Struktur. |

Beispiel

```

PROGRAM MAIN
VAR
  tSystem      : TIMESTRUCT;

```

```

    tBinary    : T_BinaryTime;
    bOverflow  : BOOL;
END_VAR

tSystem.wYear:=2022;
tSystem.wMonth:=9;
tSystem.wDay:=16;
tSystem.wHour:=12;
tSystem.wMinute:=52;
tSystem.wSecond:=4;
tSystem.wMilliseconds:=500;
tBinary:=SystemTime_To_BinaryTime(in:=tSystem, bOverflow=>bOverflow);
(* Returns tBinary.timeOfDay = TOD#12:52:04.500 and tBinary.day = 14138 *)

tBinary.day:=1;
tBinary.timeOfDay:=TOD#01:30:15.500;
tSystem:=BinaryTime_To_SystemTime(in:=tBinary, bOverflow=>bOverflow);(* Returns system-
time date: 1984-01-02 and time: 01h30m15s500ms*)

```

6.4.53 SystemTime_To_UTCtime

Konvertiert die Zeitinformation des Typs [TIMESTRUCT](#) [[454](#)] (System-Time-Format) in die Zeitinformation des Typs [T_UTCtime](#) [[479](#)].

Namensraum: [Tc3_Collections](#) [[96](#)]

Bibliothek: [Tc3_Collections](#) (Tc3_Collections.compiled-library)

Syntax

```

FUNCTION SystemTime_To_UTCtime : T_UTCtime
VAR_INPUT
    in                : TIMESTRUCT;
    wMicroseconds     : WORD(0..999);
    wNanoseconds      : WORD(0..999);
    bLeapSecondsKnown : BOOL;
    bClockFailure      : BOOL;
    bClockNotSynchronized : BOOL;
    eAccuracy          : E_UTCtimeAccuracy;
END_VAR

```

Eingänge

| Name | Typ | Beschreibung |
|-----------------------|---|--|
| in | TIMESTRUCT [454] | Die zu konvertierende Zeitinformation als System-Time-Struktur. |
| wMicroseconds | WORD(0..999) | Bruchteil der Millisekunde in Mikrosekunden: „0..999“. |
| wNanoseconds | WORD(0..999) | Bruchteil der Mikrosekunde in Nanosekunden: „0..999“. |
| bLeapSecondsKnown | BOOL | Qualität der Schaltsekunden:
„FALSE“ := Unbekannt;
„TRUE“ := Bekannt; |
| bClockFailure | BOOL | Qualität des Zeitgebers:
„FALSE“ := Gut;
„TRUE“ := Fehlerhaft; |
| bClockNotSynchronized | BOOL | Qualität der Uhrzeitsynchronisierung:
„FALSE“ := Synchron;
„TRUE“ := Nicht synchron; |
| eAccuracy | E_UTCtimeAccuracy [449] | Genauigkeit des Bruchteils der Sekunde als Aufzählungstyp. |

Rückgabewert

| Name | Typ | Beschreibung |
|-----------------------|-------------------------------------|---|
| SystemTime_To_UtcTime | T_UtcTime [▶ 479] | Konvertierte Zeitinformation als UTC-Time-Struktur. |

Beispiel

Die Zeitinformation: 2018-03-20-11:33:05.125000000 wird mit der max. Genauigkeit von 24 Bits (max. Auflösung ~60ns) konvertiert. Die Komponenten der Systemzeit haben folgende Werte:

wYear := 2018, wMonth := 3, wDay := 20, wHour := 11, wMinute := 33, wSecond := 5, wMilliseconds := 125, wDayOfWeek := 2.

```
METHOD UtcTime_ToFrom_SystemTime : BOOL
```

```
VAR
```

```
  tSystem      : TIMESTRUCT;
  sSystem      : STRING;
  tUtc         : T_UtcTime;
  sUtc         : STRING;
  wMicroseconds : WORD;
  wNanoseconds : WORD;
```

```
END_VAR
```

```
tSystem.wYear := 2000;
```

```
tSystem.wMonth := 1;
```

```
tSystem.wDay := 1;
```

```
tSystem.wHour := 0;
```

```
tSystem.wMinute := 0;
```

```
tSystem.wSecond := 0;
```

```
tSystem.wMilliseconds := 0;
```

```
tSystem.wDayOfWeek := 0;
```

```
sSystem := SystemTime_To_String(tSystem); (* sSystem := '2000-01-01-00:00:00:000' *)
```

```
tUtc := SystemTime_To_UtcTime(tSystem, 0, 0, FALSE, FALSE, TRUE, E_UtcTimeAccuracy.Unspecified);
```

```
sUtc := UtcTime_To_String(tUtc); (* sUtc := 'UT#2000-01-01-00:00:00:000000000|001|31' *)
```

```
IF tUtc.quality.ClockNotSynchronized THEN
```

```
  tUtc.secondsSinceEpoch := DT#2018-03-20-11:33:05;
```

```
  tUtc.fractionOfSecond := LTIME_TO_UTCtimeFractionOfSecond( LTIME#125MS0US0NS );
```

```
  tUtc.quality.LeapSecondsKnown := FALSE;
```

```
  tUtc.quality.ClockFailure := FALSE;
```

```
  tUtc.quality.ClockNotSynchronized := FALSE;
```

```
  Accuracy_To_UTCtimeQualityAccuracy( E_UtcTimeAccuracy._03,
```

```
    bAccuracy0=>tUtc.quality.Accuracy0,
```

```
    bAccuracy1=>tUtc.quality.Accuracy1,
```

```
    bAccuracy2=>tUtc.quality.Accuracy2,
```

```
    bAccuracy3=>tUtc.quality.Accuracy3,
```

```
    bAccuracy4=>tUtc.quality.Accuracy4 );
```

```
  sUtc := UtcTime_To_String(tUtc); (* sUtc := 'UT#2018-03-20-11:33:05.125000000|000|3' *)
```

```
  tSystem := UtcTime_To_SystemTime(tUtc, wMicroseconds=>wMicroseconds, wNanoseconds=>wNanoseconds)
```

```
;
```

```
  sSystem := SystemTime_To_String(tSystem); (* sSystem := '2018-03-20-11:33:05.125' *)
```

```
END_IF
```

```
UtcTime_ToFrom_SystemTime := TRUE;
```

6.4.54 UINT24

Initialisiert eine vorzeichenlose 24-Bit-Zahl des Typs: T_UINT24.

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

```
FUNCTION UINT24 : T_UINT24
```

```
VAR_INPUT
```

```
  Hi : BYTE;
```

```
  Mi : BYTE;
```

```
  Lo : BYTE;
```

```
END_VAR
```


 Eingänge

| Name | Typ | Beschreibung |
|------|------|--|
| Hi | BYTE | Obersten 8 Bits, die gesetzt werden sollen. |
| Mi | BYTE | Mittleren 8 Bits, die gesetzt werden sollen. |
| Lo | BYTE | Untersten 8 Bits, die gesetzt werden sollen. |

 Rückgabewert

| Name | Typ | Beschreibung |
|--------|------------------------------------|--------------|
| UINT24 | T_UINT24 [▶ 479] | 24-Bit-Zahl. |

Beispiel

```
METHOD FINAL Sample_UINT24 : BOOL
VAR
    nValue: T_UINT24;
END_VAR
nValue:=UINT24(16#12, 16#34, 16#56);(* nValue:=16#123456 *)
Sample_UINT24:=TRUE;
```

6.4.55 UINT128

Initialisiert eine vorzeichenlose 128-Bit-Zahl des Typs: T_UINT128.

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

```
FUNCTION UINT128 : T_UINT128
VAR_INPUT
    Hi : LWORD;
    Lo : LWORD;
END_VAR
```

 Eingänge

| Name | Typ | Beschreibung |
|------|-------|---|
| Hi | LWORD | Obersten 64 Bits, die gesetzt werden sollen. |
| Lo | LWORD | Untersten 64 Bits, die gesetzt werden sollen. |

 Rückgabewert

| Name | Typ | Beschreibung |
|---------|-------------------------------------|---------------|
| UINT128 | T_UINT128 [▶ 479] | 128-Bit-Zahl. |

Beispiel

```
METHOD FINAL Sample_UINT128 : BOOL
VAR
    nValue: T_UINT128;
END_VAR
nValue:=UINT128(16#1122334455667788, 16#0000000000000000);
(* nValue:=16#1122334455667788_0000000000000000 *)
Sample_UINT128:=TRUE;
```

6.4.56 UtcTime

Initialisiert die Datenelemente der Struktur: T_UtcTime.

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

```
FUNCTION UtcTime : T_UtcTime
VAR_INPUT
    secondSinceEpoch : DT;
    fractionOfSecond : T_UINT24;
    quality            : T_UtcTimeQuality;
END_VAR
```

Eingänge

| Name | Typ | Beschreibung |
|------------------|--|--|
| secondSinceEpoch | DT | Anzahl der vergangenen/ganzen Sekunden seit dem 1 Januar 1970. |
| fractionOfSecond | T_UINT24 [▶ 479] | Sekundenbruchteil codiert als 24-Bit-Zahl. |
| quality | T_UtcTimeQuality [▶ 480] | Zusätzliche Information über die Qualität und die Genauigkeit der Zeitinformation. |

Rückgabewert

| Name | Typ | Beschreibung |
|---------|---|--|
| UtcTime | T_UtcTime [▶ 479] | Struktur mit initialisierten Datenelementen. |

Beispiel

```
METHOD FINAL Sample_UtcTime : BOOL
VAR
    t : T_UtcTime;
END_VAR
```

```
t:=UtcTime(secondSinceEpoch:=DT#2023-01-01-00:00:00.000, fractionOfSecond:=UINT24(0, 0, 0), quality:=UtcTimeQuality(FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE));
Sample_UtcTime:=TRUE;
```

6.4.57 UtcTime_To_LWORD

Konvertiert die Komponenten des strukturierten Typs [T_UtcTime](#) [[▶ 479](#)] in die Binärdaten des Typs LWORD. Die Kodierung der Komponenten im LWORD entspricht der TwinCAT [UTC-Time LWORD-Formatspezifikation](#) [[▶ 682](#)].

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
FUNCTION UtcTime_To_LWORD : LWORD
VAR_INPUT
    in : T_UtcTime;
END_VAR
```

Eingänge

| Name | Typ | Beschreibung |
|------|---|--|
| in | T_UtcTime [▶ 479] | Die zu konvertierende Zeitinformation als UTC-Time-Struktur. |

Rückgabewert

| Name | Typ | Beschreibung |
|------------------|-------|---|
| UtcTime_To_LWORD | LWORD | Konvertierte Zeitinformation als UTC-Time-Binärdaten. |

Beispiel

```
METHOD UtcTime_ToFrom_LWORD : BOOL
VAR
    tValue : T_UtcTime;
    sValue : STRING;
END_VAR

tValue := LWORD_To_UtcTime(0);
sValue := UtcTime_To_String(tValue); (* sValue := 'UT#1970-01-01-00:00:00.000000000|000|0' *)

IF UtcTime_To_LWORD(tValue) = 0 THEN
    tValue.secondSinceEpoch := DT#2000-01-01-00:00:00;
    tValue.fractionOfSecond := LTIME_To_UtcTimeFractionOfSecond( LTIME#0MS );
    tValue.quality.LeapSecondsKnown := FALSE;
    tValue.quality.ClockFailure := FALSE;
    tValue.quality.ClockNotSynchronized := TRUE;
    Accuracy_To_UtcTimeQualityAccuracy( E_UtcTimeAccuracy._03,
                                         bAccuracy0=>tValue.quality.Accuracy0,
                                         bAccuracy1=>tValue.quality.Accuracy1,
                                         bAccuracy2=>tValue.quality.Accuracy2,
                                         bAccuracy3=>tValue.quality.Accuracy3,
                                         bAccuracy4=>tValue.quality.Accuracy4 );
    sValue := UtcTime_To_String(tValue); (* sValue := 'UT#2000-01-01-00:00:00.000000000|001|3' *)
END_IF

UtcTime_ToFrom_LWORD := TRUE;
```

6.4.58 UtcTime_To_String

Konvertiert die Komponenten des strukturierten Typs [T_UtcTime \[▶ 479\]](#) in einen formatierten UTC-Time-String. Die Zeichenfolge des resultierenden Strings hat folgenden Aufbau: 'UT#YYYY-MM-DD-hh:mm:ss.nnnnnnnn[LFC|A[A]'. Sie entspricht der TwinCAT [UTC-Time-String-Formatspezifikation \[▶ 686\]](#).

Namensraum: [Tc3_Collections \[▶ 96\]](#)

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
FUNCTION UtcTime_To_String : STRING(39)
VAR_INPUT
    in : T_UtcTime;
END_VAR
VAR_OUTPUT
    bError : BOOL;
END_VAR
```

 **Eingänge**

| Name | Typ | Beschreibung |
|------|-----------------------------------|--|
| in | T_UtcTime [▶ 479] | Die zu konvertierende Zeitinformation als UTC-Time-Struktur. |

 **Ausgänge**

| Name | Typ | Beschreibung |
|--------|------|--|
| bError | BOOL | Der Ausgang ist „TRUE“ wenn der Eingangswert nicht fehlerfrei konvertiert werden konnte. |

 **Rückgabewert**

| Name | Typ | Beschreibung |
|-------------------|------------|---|
| UtcTime_To_String | STRING(39) | Konvertierte Zeitinformation als UTC-Time-String. |

Beispiel

'UT#2017-01-16-11:22:33.750000000|001|24' bedeutet Datum: „2017-01-16“, Uhrzeit: „11:22:33“ und „750“ ms, Schaltsekunden unbekannt, Zeitgeber hat keinen Fehler, Uhrzeit ist nicht synchronisiert.

```
METHOD UtcTime_ToFrom_String : BOOL
VAR_INPUT
    fbIED : REFERENCE TO FB_IED;
END_VAR
VAR
    tValue : T_UtcTime;
    sValue : STRING;
END_VAR

IF NOT __ISVALIDREF(fbIED) THEN
    RETURN;
END_IF

tValue := String_To_UtcTime('UT#1970-01-01-00:00:00.000000000|000|0');
sValue := UtcTime_To_String(tValue); (* sValue := 'UT#1970-01-01-00:00:00.000000000|000|0' *)
IF fbIED.Relay.LLN0.Beh.t.sValue <> sValue THEN
    tValue := fbIED.Relay.LLN0.Beh.t.tValue;
END_IF

tValue := String_To_UtcTime('UT#2018-03-21-14:56:55.125000000|010|3');
(* ClockFailure := 1, accuracy := 3 bits *)
sValue := UtcTime_To_String(tValue); (* sValue := 'UT#2018-03-21-14:56:55.125000000|010|3' *)
IF fbIED.Relay.LLN0.Health.t.sValue <> sValue THEN
    tValue := fbIED.Relay.LLN0.Health.t.tValue;
END_IF

tValue := String_To_UtcTime('UT#2017-01-16-11:22:33.750000000|001|24');
(* ClockNotSynchronized := 1, accuracy := 24 bits *)
sValue := UtcTime_To_String(tValue); (* sValue := 'UT#2017-01-16-11:22:33.750000000|001|24' *)
IF fbIED.Relay.LLN0.Mod_.t.sValue <> sValue THEN
    tValue := fbIED.Relay.LLN0.Mod_.t.tValue;
END_IF

UtcTime_ToFrom_String := TRUE;
```

6.4.59 UtcTime_To_SystemTime

Konvertiert die Zeitinformation des Typs T_UtcTime [▶ 479] in die Zeitinformation des Typs TIMESTRUCT [▶ 454] (System-Time-Format).

Namensraum: Tc3_Collections [▶ 96]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
FUNCTION UtcTime_To_SystemTime : TIMESTRUCT
VAR_INPUT
    in          : T_UtcTime;
END_VAR
VAR_OUTPUT
    wMicroseconds : WORD(0..999);
    wNanoseconds  : WORD(0..999);
    bLeapSecondsKnown : BOOL;
    bClockFailure   : BOOL;
    bClockNotSynchronized : BOOL;
    eAccuracy       : E_UtcTimeAccuracy;
END_VAR
```

👉 Eingänge

| Name | Typ | Beschreibung |
|------|--------------------------|--|
| in | <u>T_UtcTime</u> [▶ 479] | Die zu konvertierende Zeitinformation als UTC-Time-Struktur. |

 **Ausgänge**

| Name | Typ | Beschreibung |
|-----------------------|---|--|
| wMicroseconds | WORD(0..999) | Optionaler Ausgangsparameter. Bruchteil der Millisekunde in Mikrosekunden. |
| wNanoseconds | WORD(0..999) | Optionaler Ausgangsparameter. Bruchteil der Mikrosekunde in Nanosekunden. |
| bLeapSecondsKnown | BOOL | Optionaler Ausgangsparameter. Qualität der Schaltsekunden. |
| bClockFailure | BOOL | Optionaler Ausgangsparameter. Qualität des Zeitgebers. |
| bClockNotSynchronized | BOOL | Optionaler Ausgangsparameter. Qualität der Uhrzeitsynchronisation. |
| eAccuracy | E_UtcTimeAccuracy [▶ 449] | Genauigkeit des Bruchteils der Sekunde. Dieser Wert bestimmt die max. Anzahl der signifikanten Bits der T_UtcTime [▶ 479].fractionOfSecond-Komponente. |

 **Rückgabewert**

| Name | Typ | Beschreibung |
|-----------------------|--|--|
| UtcTime_To_SystemTime | TIMESTRUCT [▶ 454] | Konvertierte Zeitinformation als System-Time-Struktur. |

Beispiel

Die Zeitinformation: 2018-03-20-11:33:05.125000000 wird mit der max. Genauigkeit von 24 Bits (max. Auflösung ~60ns) konvertiert. Die Komponenten der Systemzeit haben folgende Werte:

wYear := 2018, wMonth := 3, wDay := 20, wHour := 11, wMinute := 33, wSecond := 5, wMilliseconds := 125, wDayOfWeek := 2.

```
METHOD UtcTime_ToFrom_SystemTime : BOOL
VAR
    tSystem      : TIMESTRUCT;
    sSystem      : STRING;
    tUtc         : T_UtcTime;
    sUtc         : STRING;
    wMicroseconds : WORD;
    wNanoseconds  : WORD;
END_VAR

tSystem.wYear := 2000;
tSystem.wMonth := 1;
tSystem.wDay := 1;
tSystem.wHour := 0;
tSystem.wMinute := 0;
tSystem.wSecond := 0;
tSystem.wMilliseconds := 0;
tSystem.wDayOfWeek := 0;
sSystem := SystemTime_To_String(tSystem); (* sSystem := '2000-01-01-00:00:00:000' *)
tUtc := SystemTime_To_UtcTime(tSystem, 0, 0, FALSE, FALSE, TRUE, E_UtcTimeAccuracy.Unspecified);
sUtc := UtcTime_To_String(tUtc); (* sUtc := 'UT#2000-01-01-00:00:00:000000000|001|31' *)

IF tUtc.quality.ClockNotSynchronized THEN
    tUtc.secondsSinceEpoch := DT#2018-03-20-11:33:05;
    tUtc.fractionOfSecond := LTIME_TO_UtcTimeFractionOfSecond( LTIME#125MS0USONS );
    tUtc.quality.LeapSecondsKnown := FALSE;
    tUtc.quality.ClockFailure := FALSE;
    tUtc.quality.ClockNotSynchronized := FALSE;
    Accuracy_To_UtcTimeQualityAccuracy( E_UtcTimeAccuracy._03,
                                        bAccuracy0=>tUtc.quality.Accuracy0,
                                        bAccuracy1=>tUtc.quality.Accuracy1,
                                        bAccuracy2=>tUtc.quality.Accuracy2,
                                        bAccuracy3=>tUtc.quality.Accuracy3,
                                        bAccuracy4=>tUtc.quality.Accuracy4 );
END IF
```

```

    sUtc := UtcTime_To_String(tUtc); (* sUtc := 'UT#2018-03-20-11:33:05.125000000|000|3' *)
    tSystem := UtcTime_To_SystemTime(tUtc, wMicroseconds=>wMicroseconds, wNanoseconds=>wNanoseconds)
;
    sSystem := SystemTime_To_String(tSystem); (* sSystem := '2018-03-20-11:33:05.125' *)
END_IF

UtcTime_ToFrom_SystemTime := TRUE;

```

6.4.60 UtcTimeFractionOfSecond_To_LTIME

Konvertiert den Bruchteil einer Sekunde des Typs [T_UINT24](#) [▶ 479] in den Typ LTIME. Die zu konvertierenden Binärdaten entsprechen dem Typ der [T_UtcTime](#) [▶ 479].fractionOfSecond-Komponente. Die 24 Bits müssen auf besondere Weise codiert sein. Siehe: [Codierung der UTC-Time.fractionOfSecond0..23-Bits](#) [▶ 682]. Die [T_UtcTime](#) [▶ 479].Quality.Accuracy0..4-Bit-Komponenten bestimmen die max. Anzahl der signifikanten fractionOfSecond-Bits und die max. Auflösung die bei der Konvertierung erreicht werden kann (~60ns).

Namensraum: [Tc3_Collections](#) [▶ 96]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```

FUNCTION UtcTimeFractionOfSecond_To_LTIME : LTIME
VAR_INPUT
    in      : T_UINT24;
END_VAR
VAR_OUTPUT
    nanosec : LWORD;
    attosec : LWORD;
END_VAR

```

Eingänge

| Name | Typ | Beschreibung |
|------|----------------------------------|---|
| in | T_UINT24 [▶ 479] | Der zu konvertierende Bruchteil der Sekunde als Binärdaten. |

Ausgänge

| Name | Typ | Beschreibung |
|---------|-------|--|
| nanosec | LWORD | Optionaler Ausgangsparameter. Bruchteil der Sekunde in Nanosekunden. |
| attosec | LWORD | Optionaler Ausgangsparameter. Bruchteil der Sekunde in Attosekunden. |

Rückgabewert

| Name | Typ | Beschreibung |
|----------------------------------|-------|--|
| UtcTimeFractionOfSecond_To_LTIME | LTIME | Konvertierter Bruchteil einer Sekunde in Nanosekunden. |

Beispiel

```

METHOD UtcTimeFractionOfSecond_ToFrom_LTIME : BOOL
VAR
    tValue      : T_UtcTime;
    sValue      : STRING;
    tFos        : LTIME;
    nanosec     : LWORD;
END_VAR

tValue.quality := String_To_UtcTimeQuality('UQ#000|24');
tValue.secondSinceEpoch := DT#2018-03-28-12:13:14;

tValue.fractionOfSecond := LTIME_To_UtcTimeFractionOfSecond(LTIME#0NS);

```

```
(* tValue.fractionOfSecond := [16#00, 16#00, 16#00], 0 significant bits *)
sValue := UtcTime_To_String(tValue);(* sValue := 'UT#2018-03-28-12:13:14.000000000|000|24' *)
tFos := UtcTimeFractionOfSecond_To_LTIME(tValue.fractionOfSecond, nanosec=>nanosec);
(* tFos := LTIME#0ns, nanosec := 0 *)

tValue.fractionOfSecond := LTIME_To_UtcTimeFractionOfSecond(LTIME#750MS);
(* tValue.fractionOfSecond := [16#03, 16#00, 16#00], 3 significant bits *)
sValue := UtcTime_To_String(tValue);(* sValue := 'UT#2018-03-28-12:13:14.750000000|000|24' *)
tFos := UtcTimeFractionOfSecond_To_LTIME(tValue.fractionOfSecond, nanosec=>nanosec);
(* tFos := LTIME#750ms, nanosec := 750000000 *)

tValue.fractionOfSecond := LTIME_To_UtcTimeFractionOfSecond(LTIME#968MS750US);
(* tValue.fractionOfSecond := [16#1F, 16#00, 16#00], 5 significant bits *)
sValue := UtcTime_To_String(tValue);(* sValue := 'UT#2018-03-28-12:13:14.968750000|000|24' *)
tFos := UtcTimeFractionOfSecond_To_LTIME(tValue.fractionOfSecond, nanosec=>nanosec);
(* tFos := LTIME#968ms750us, nanosec := 968750000 *)

tValue.fractionOfSecond := LTIME_To_UtcTimeFractionOfSecond(LTIME#998MS46US875NS);
(* tValue.fractionOfSecond := [16#FF, 16#01, 16#00], 9 significant bits *)
sValue := UtcTime_To_String(tValue);(* sValue := 'UT#2018-03-28-12:13:14.998046875|000|24' *)
tFos := UtcTimeFractionOfSecond_To_LTIME(tValue.fractionOfSecond, nanosec=>nanosec);
(* tFos := LTIME#998ms46us875ns, nanosec := 998046875 *)

tValue.fractionOfSecond := LTIME_To_UtcTimeFractionOfSecond(LTIME#999MS999US999NS);
(* tValue.fractionOfSecond := [16#FF, 16#FF, 16#FF], 24 significant bits *)
sValue := UtcTime_To_String(tValue);(* sValue := 'UT#2018-03-28-12:13:14.999999940|000|24' *)
tFos := UtcTimeFractionOfSecond_To_LTIME(tValue.fractionOfSecond, nanosec=>nanosec);
(* tFos := LTIME#999ms999us940ns, nanosec := 999999940 *)

UtcTimeFractionOfSecond_ToFrom_LTIME := TRUE;
```

6.4.61 UtcTimelsNull

Diese Funktion liefert „TRUE“ wenn alle Komponenten des strukturierten Typs T_UtcTime [\[▶ 479\]](#) „0“ sind.

Namensraum: Tc3_Collections [\[▶ 96\]](#)

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
FUNCTION UtcTimeIsNull : BOOL
VAR_INPUT
    in : T_UtcTime;
END_VAR
```

Eingänge

| Name | Typ | Beschreibung |
|------|--|--|
| in | <u>T_UtcTime</u> [▶ 479] | UTC-Time-Wert der überprüft werden soll. |

Rückgabewert

| Name | Typ | Beschreibung |
|---------------|------|--|
| UtcTimelsNull | BOOL | TRUE wenn alle Komponente von UTC-Time „0“ sind. |

Beispiel

```
PROGRAM P_UtcTimeIsNull
VAR
    t      : T_UtcTime;
    bNull : BOOL;
END_VAR

t      := SEL( bNull, Null_TimeStamp, String_To_UtcTime('UT#2018-03-21-15:58:03.125000000|000|3') );
bNull := UtcTimeIsNull(t);
```

6.4.62 UtcTimeQuality

Initialisiert die Datenelemente der Struktur: T_UtcTimeQuality.

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

```
FUNCTION UtcTimeQuality : T_UtcTimeQuality
VAR_INPUT
    LeapSecondsKnown      : BOOL;
    ClockFailure          : BOOL;
    ClockNotSynchronized : BOOL;
    Accuracy0             : BOOL;
    Accuracy1             : BOOL;
    Accuracy2             : BOOL;
    Accuracy3             : BOOL;
    Accuracy4             : BOOL;
END_VAR
```

Eingänge

| Name | Typ | Beschreibung |
|----------------------|------|---|
| LeapSecondsKnown | BOOL | Datenelement „LeapSecondsKnown“, das gesetzt werden soll. |
| ClockFailure | BOOL | Datenelement „ClockFailure“, das gesetzt werden soll. |
| ClockNotSynchronized | BOOL | Datenelement „ClockNotSynchronized“, das gesetzt werden soll. |
| Accuracy0 | BOOL | Datenelement „Accuracy0“, das gesetzt werden soll. |
| Accuracy1 | BOOL | Datenelement „Accuracy1“, das gesetzt werden soll. |
| Accuracy2 | BOOL | Datenelement „Accuracy2“, das gesetzt werden soll. |
| Accuracy3 | BOOL | Datenelement „Accuracy3“, das gesetzt werden soll. |
| Accuracy4 | BOOL | Datenelement „Accuracy4“, das gesetzt werden soll. |

Rückgabewert

| Name | Typ | Beschreibung |
|----------------|--|--|
| UtcTimeQuality | T_UtcTimeQuality [▶ 480] | Struktur mit initialisierten Datenelementen. |

Beispiel

```
METHOD FINAL Sample_UtcTimeQuality : BOOL
VAR
    q: T_UtcTimeQuality;
END_VAR

q:=UtcTimeQuality(FALSE, FALSE, TRUE, FALSE, FALSE, FALSE, FALSE, FALSE);
(* q.ClockNotSynchronized:=TRUE, all other elements: FALSE *)
Sample_UtcTimeQuality:=TRUE;
```

6.4.63 UtcTimeQuality_To_BYTE

Konvertiert die Komponenten des strukturierten Typs [T_UtcTimeQuality](#) [[▶ 480](#)] in die Binärdaten des Typs BYTE. Die Kodierung der Komponenten im BYTE entspricht der TwinCAT [UTC-Time-Quality-BYTE-Formatspezifikation](#) [[▶ 684](#)].

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
FUNCTION UtcTimeQuality_To_BYTE : BYTE
VAR_INPUT
    in : T_UtcTimeQuality;
END_VAR
```


 Eingänge

| Name | Typ | Beschreibung |
|------|--|--|
| in | T_UtcTimeQuality [▶ 480] | Die zu konvertierende Information als UTC-Time-Quality-Struktur. |

 Rückgabewert

| Name | Typ | Beschreibung |
|------------------------|------|---|
| UtcTimeQuality_To_BYTE | BYTE | Konvertierte Information als UTC-Time-Quality-Binärdaten. |

Beispiel

```
METHOD UtcTimeQuality_ToFrom_Byte : BOOL
VAR_INPUT
    fbIED      : REFERENCE TO FB_IED;
END_VAR
VAR
    Quality    : T_UtcTimeQuality;
    sQuality   : STRING;
    u8         : BYTE;
END_VAR

IF NOT __ISVALIDREF(fbIED) THEN
    RETURN;
END_IF

Quality := Byte_To_UtcTimeQuality(0);
u8 := UtcTimeQuality_To_Byte(Quality); (* u8 := 2#00000000 => 0 bits accuracy *)
sQuality := UtcTimeQuality_To_String(Quality); (* sQuality := 'UQ#000|00' *)
IF UtcTimeQuality_To_Byte(fbIED.Relay.LLNO.Beh.t.Quality) <> u8 THEN
    Quality := fbIED.Relay.LLNO.Beh.t.Quality;
END_IF

Quality := String_To_UtcTimeQuality('UQ#000|03');
u8 := UtcTimeQuality_To_Byte(Quality); (* u8 := 2#11000000 => 3 bits accuracy *)
sQuality := UtcTimeQuality_To_String(Quality); (* sQuality := 'UQ#000|03' *)
IF UtcTimeQuality_To_Byte(fbIED.Relay.LLNO.Health.t.Quality) <> u8 THEN
    Quality := fbIED.Relay.LLNO.Health.t.Quality;
END_IF

Quality := String_To_UtcTimeQuality('UQ#001|05');
u8 := UtcTimeQuality_To_Byte(Quality);
(* u8 := 2#10100100 => 5 bits accuracy, ClockNotSynchronized := 1 *)
sQuality := UtcTimeQuality_To_String(Quality); (* sQuality := 'UQ#001|05' *)
IF UtcTimeQuality_To_Byte(fbIED.Relay.LLNO.Mod_.t.Quality) <> u8 THEN
    Quality := fbIED.Relay.LLNO.Mod_.t.Quality;
END_IF

Quality := String_To_UtcTimeQuality('UQ#000|31');
u8 := UtcTimeQuality_To_Byte(Quality); (* u8 := 2#11111000 => unspecified accuracy *)
sQuality := UtcTimeQuality_To_String(Quality); (* sQuality := 'UQ#000|31' *)
IF UtcTimeQuality_To_Byte(fbIED.Relay.MMXU1.phV.phsA.t.Quality) <> u8 THEN
    Quality := fbIED.Relay.MMXU1.phV.phsA.t.Quality;
END_IF

UtcTimeQuality_ToFrom_Byte := TRUE;
```

6.4.64 UtcTimeQuality_To_String

Konvertiert die Komponenten des strukturierten Typs [T_UtcTimeQuality](#) [[▶ 480](#)] in einen formatierten UTC-Time-Quality-String. Die Zeichenfolge des resultierenden Strings hat folgenden Aufbau: 'UQ#LFC|A[A]'. Sie entspricht der TwinCAT [UTC-Time-Quality-String-Formatspezifikation](#) [[▶ 685](#)].

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```

FUNCTION UtcTimeQuality_To_String : STRING(9)
VAR_INPUT
    in      : T_UtcTimeQuality;
END_VAR
VAR_OUTPUT
    bError  : BOOL;
END_VAR

```

Eingänge

| Name | Typ | Beschreibung |
|------|--------------------------|--|
| in | T_UtcTimeQuality [► 480] | Die zu konvertierende Information als UTC-Time-Quality-Struktur. |

Ausgänge

| Name | Typ | Beschreibung |
|--------|------|--|
| bError | BOOL | Der Ausgang ist „TRUE“ wenn der Eingangswert nicht fehlerfrei konvertiert werden konnte. |

Rückgabewert

| Name | Typ | Beschreibung |
|--------------------------|-----------|--|
| UtcTimeQuality_To_String | STRING(9) | Konvertiert Information als UTC-Time-Quality-String. |

Beispiel

Der Rückgabewert: 'UQ#001|3' bedeutet, dass Schaltsekunden nicht bekannt sind. Zeitgeber hat keinen Fehler und ist nicht synchronisiert, die Auflösung der Zeitinformation des Bruchteils der Sekunde beträgt 3 Bits.

```

METHOD UtcTimeQuality_ToFrom_String : BOOL
VAR_INPUT
    fbIED      : REFERENCE TO FB_IED;
END_VAR
VAR
    Quality    : T_UtcTimeQuality;
    sQuality   : STRING;
END_VAR

IF NOT __ISVALIDREF(fbIED) THEN
    RETURN;
END_IF

Quality := String_To_UtcTimeQuality('UQ#000|0');
sQuality := UtcTimeQuality_To_String(Quality); (* sQuality := 'UQ#000|0' *)
IF fbIED.Relay.LLN0.Beh.t.sQuality <> sQuality THEN
    Quality := fbIED.Relay.LLN0.Beh.t.Quality;
END_IF

Quality := String_To_UtcTimeQuality('UQ#001|3');
sQuality := UtcTimeQuality_To_String(Quality); (* sQuality := 'UQ#001|3' *)
IF fbIED.Relay.LLN0.Health.t.sQuality <> sQuality THEN
    Quality := fbIED.Relay.LLN0.Health.t.Quality;
END_IF

Quality := String_To_UtcTimeQuality('UQ#000|31');
sQuality := UtcTimeQuality_To_String(Quality); (* sQuality := 'UQ#000|31' *)
IF fbIED.Relay.LLN0.Mod.t.sQuality <> sQuality THEN
    Quality := fbIED.Relay.LLN0.Mod.t.Quality;
END_IF

UtcTimeQuality_ToFrom_String := TRUE;

```

6.4.65 UtcTimeQualityAccuracy_To_Accuracy

Konvertiert die Genauigkeit der Zeitinformation des strukturierten Typs [T_UtcTimeQuality](#) [▶ 480] in den Aufzählungstyp [E_UtcTimeAccuracy](#) [▶ 449]. Es werden nur die Accuracy0..4 Bit-Komponenten konvertiert. Diese Variablen bestimmen die max. Anzahl der signifikanten Bits in der [T_UtcTime](#) [▶ 479].fractionOfSecond Komponente.

Namensraum: [Tc3_Collections](#) [▶ 96]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
FUNCTION UtcTimeQualityAccuracy_To_Accuracy : E_UtcTimeAccuracy
VAR_INPUT
    in : T_UtcTimeQuality;
END_VAR
```

Eingänge

| Name | Typ | Beschreibung |
|------|--|---|
| in | T_UtcTimeQuality [▶ 480] | Die zu konvertierende Information als UTC-Time-Quality.Accuracy0..4-Bit-Komponenten.
Siehe: Kodierung der Accuracy0..4-Bits [▶ 683]. |

Rückgabewert

| Name | Typ | Beschreibung |
|------------------------------------|---|---|
| UtcTimeQualityAccuracy_To_Accuracy | E_UtcTimeAccuracy [▶ 449] | Konvertierte Information als Accuracy-Aufzählungstyp. |

Beispiel

```
METHOD UtcTimeQualityAccuracy_ToFrom_Accuracy : BOOL
VAR_INPUT
    fbIED : REFERENCE TO FB_IED;
END_VAR
VAR
    Quality : T_UtcTimeQuality;
    sQuality : STRING;
END_VAR

IF NOT __ISVALIDREF(fbIED) THEN
    RETURN;
END_IF

Quality.LeapSecondsKnown := FALSE;
Quality.ClockFailure := FALSE;
Quality.ClockNotSynchronized := FALSE;
Accuracy_To_UtcTimeQualityAccuracy( E_UtcTimeAccuracy.Null, (* Null bits accuracy *)
    bAccuracy0=>Quality.Accuracy0,
    bAccuracy1=>Quality.Accuracy1,
    bAccuracy2=>Quality.Accuracy2,
    bAccuracy3=>Quality.Accuracy3,
    bAccuracy4=>Quality.Accuracy4 );
sQuality := UtcTimeQuality_To_String(Quality); (* sQuality := 'UQ#000|0' *)
IF fbIED.Relay.LLN0.Beh.t.eAccuracy <> UtcTimeQualityAccuracy_To_Accuracy(Quality) THEN
    Quality := fbIED.Relay.LLN0.Beh.t.Quality;
END_IF

Quality.LeapSecondsKnown := FALSE;
Quality.ClockFailure := FALSE;
Quality.ClockNotSynchronized := FALSE;
Accuracy_To_UtcTimeQualityAccuracy( E_UtcTimeAccuracy._03, (* 3 bits accuracy *)
    bAccuracy0=>Quality.Accuracy0,
    bAccuracy1=>Quality.Accuracy1,
    bAccuracy2=>Quality.Accuracy2,
    bAccuracy3=>Quality.Accuracy3,
    bAccuracy4=>Quality.Accuracy4 );
sQuality := UtcTimeQuality_To_String(Quality); (* sQuality := 'UQ#000|3' *)
```

```

IF fbIED.Relay.LLN0.Health.t.eAccuracy <> UtcTimeQualityAccuracy_To_Accuracy(Quality) THEN
    Quality := fbIED.Relay.LLN0.Health.t.Quality;
END_IF

Quality.LeapSecondsKnown := FALSE;
Quality.ClockFailure := FALSE;
Quality.ClockNotSynchronized := FALSE;
Accuracy_To_UtcTimeQualityAccuracy( E_UtcTimeAccuracy._05, (* 5 bits accuracy *)
    bAccuracy0=>Quality.Accuracy0,
    bAccuracy1=>Quality.Accuracy1,
    bAccuracy2=>Quality.Accuracy2,
    bAccuracy3=>Quality.Accuracy3,
    bAccuracy4=>Quality.Accuracy4 );
sQuality := UtcTimeQuality_To_String(Quality); (* sQuality := 'UQ#000|5' *)
IF fbIED.Relay.LLN0.Mod_.t.eAccuracy <> UtcTimeQualityAccuracy_To_Accuracy(Quality) THEN
    Quality := fbIED.Relay.LLN0.Mod_.t.Quality;
END_IF

Quality.LeapSecondsKnown := FALSE;
Quality.ClockFailure := FALSE;
Quality.ClockNotSynchronized := FALSE;
Accuracy_To_UtcTimeQualityAccuracy( E_UtcTimeAccuracy.Unspecified, (* Unspecified accuracy *)
    bAccuracy0=>Quality.Accuracy0,
    bAccuracy1=>Quality.Accuracy1,
    bAccuracy2=>Quality.Accuracy2,
    bAccuracy3=>Quality.Accuracy3,
    bAccuracy4=>Quality.Accuracy4 );
sQuality := UtcTimeQuality_To_String(Quality); (* sQuality := 'UQ#000|31' *)
IF fbIED.Relay.MMXU1.phV.phsA.t.eAccuracy <> UtcTimeQualityAccuracy_To_Accuracy(Quality) THEN
    Quality := fbIED.Relay.MMXU1.phV.phsA.t.Quality;
END_IF

UtcTimeQualityAccuracy_ToFrom_Accuracy := TRUE;

```

6.4.66 UtcTimeStamp_To_String

Konvertiert die Komponenten des strukturierten Typs [T_UtcTime](#) [► 479] in einen formatierten UTC-Zeitstempel-String. Die Zeichenfolge des resultierenden Strings hat folgenden Aufbau: 'US#YYYY-MM-DD-hh:mm:ss.nnnnnnnn':

YYYY:=Jahr

MM:=Monat

DD:=Tag

hh:=Stunde

mm:=Minuten

ss:=Sekunden

nnnnnnnn:=Nanosekunden (9 Dezimalstellen).

Die [T_UtcTime](#) [► 479].Quality-Komponente wird nicht in den String konvertiert.

Namensraum: [Tc3_Collections](#) [► 96]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```

FUNCTION UtcTimeStamp_To_String : STRING(32)
VAR_INPUT
    in      : T_UtcTime;
END_VAR
VAR_OUTPUT
    bError : BOOL;
END_VAR

```

Eingänge

| Name | Typ | Beschreibung |
|------|-----------------------------------|--|
| in | T_UtcTime [► 479] | Die zu konvertierende Zeitinformation als UTC-Time-Struktur. |

 **Ausgänge**

| Name | Typ | Beschreibung |
|--------|------|--|
| bError | BOOL | Der Ausgang ist „TRUE“ wenn der Eingangswert nicht fehlerfrei konvertiert werden konnte. |

 **Rückgabewert**

| Name | Typ | Beschreibung |
|------------------------|------------|---|
| UtcTimeStamp_To_String | STRING(32) | Konvertierte Zeitinformation als formatierter UTC-Zeitstempel-String. |

Beispiel

```
PROGRAM MAIN
VAR
    tUtc          : T_UtcTime;
    sUtc          : STRING;
    bError        : BOOL;
    secondSinceEpoch : DT;
    fractionOfSecond : T_UINT24;
END_VAR

tUtc:=String_To_UtcTimeStamp(in:='US#2022-09-15-16:12:34.750000000' , bError=>bError, secondSinceEpoch=>secondSinceEpoch, fractionOfSecond=>fractionOfSecond);
sUtc:=UtcTimeStamp_To_String(in:=tUtc, bError=>bError);
(* => sUtc = 'US#2022-09-15-16:12:34.750000000' *)

tUtc.secondSinceEpoch:=DT#2022-09-15-00:00:00;
tUtc.fractionOfSecond[0]:=0;
tUtc.fractionOfSecond[1]:=0;
tUtc.fractionOfSecond[2]:=0;
sUtc:=UtcTimeStamp_To_String(in:=tUtc, bError=>bError);
(* => sUtc = 'US#2022-09-15-00:00:00.000000000' *)
```

6.4.67 Validity_To_AcsiQualityValidity

Konvertiert den Wert des Aufzählungstyps [E_AcsiQualityValidity](#) [▶ 425] in zwei boolische Werte. Diese zwei Ausgangswerte entsprechen den Validity0- und Validity1-Bit-Komponenten des strukturierten Typs [ST_AcsiQuality](#) [▶ 463]. Siehe auch: [Kodierung der Validity0..1-Bits](#) [▶ 687].

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
FUNCTION Validity_To_AcsiQualityValidity
VAR_INPUT
    in          : E_AcsiQualityValidity;
END_VAR
VAR_OUTPUT
    bValidity0 : BOOL;
    bValidity1 : BOOL;
END_VAR
```

 **Eingänge**

| Name | Typ | Beschreibung |
|------|---|---|
| in | E_AcsiQualityValidity [▶ 425] | Die zu konvertierende Information als Aufzählungstyp. |

 **Ausgänge**

Konvertierte Information als [ST_AcsiQuality](#) [▶ 463].Validity0..1-Bit-Komponenten.

| Name | Typ | Beschreibung |
|------------|------|---------------------|
| bValidity0 | BOOL | Gültigkeit MSB-Bit. |
| bValidity1 | BOOL | Gültigkeit LSB-Bit. |

Rückgabewert

Keiner.

Beispiel

```
METHOD AcsiQualityValidity_ToFrom_Validity : BOOL
VAR_INPUT
    fbIED : REFERENCE TO FB_IED;
END_VAR
VAR
    qValue : ST_AcsiQuality;
    sValue : STRING;
END_VAR

IF NOT __ISVALIDREF(fbIED) THEN
    RETURN;
END_IF
Validity_To_AcsiQualityValidity(E_AcsiQualityValidity.Good, bValidity0=>qValue.Validity0, bValidity1
=>qValue.Validity1);
qValue.Overflow := FALSE;
qValue.OutOfRange := FALSE;
qValue.BadReference := FALSE;
qValue.Oscillatory := FALSE;
qValue.Failure := FALSE;
qValue.OldData := FALSE;
qValue.Inconsistent := FALSE;
qValue.Inaccurate := FALSE;
qValue.Source := Source_To_AcsiQualitySource(E_AcsiQualitySource.Process);
qValue.Test := FALSE;
qValue.OperatorBlocked := FALSE;
sValue := AcsiQuality_To_String(qValue); (* sValue := 'Q#00|00000000|0|00' *)

IF fbIED.Relay.LLN0.Beh.q.eValidity <> AcsiQualityValidity_To_Validity(qValue) THEN
    qValue := fbIED.Relay.LLN0.Beh.q.qValue;
END_IF

AcsiQualityValidity_ToFrom_Validity := TRUE;
```

6.4.68 WORD_To_AcsiQuality

Konvertiert die Binärdaten des Typs WORD in die Komponenten des strukturierten Typs ST_AcsiQuality [▶ 463]. Die Kodierung der Komponenten im WORD entspricht der TwinCAT Quality-WORD-Formatspezifikation [▶ 688].

Namensraum: Tc3_Acsi [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)


Syntax

```
FUNCTION Word_To_AcsiQuality : ST_AcsiQuality
VAR_INPUT
    in : WORD(0..8191);
END_VAR
```

Eingänge

| Name | Typ | Beschreibung |
|------|---------------|---|
| in | WORD(0..8191) | Die zu konvertierende Information als Quality-Binärdaten. |

 Rückgabewert

| Name | Typ | Beschreibung |
|---------------------|--|--|
| WORD_To_AcsiQuality | ST_AcsiQuality  463 | Konvertierte Information als Quality-Struktur. |

Beispiel

```

METHOD AcsiQuality_ToFrom_WORD : BOOL
VAR_INPUT
    fbIED : REFERENCE TO FB_IED;
END_VAR
VAR
    qValue : ST_AcsiQuality;
    sValue : STRING;
    nValue : WORD;
END_VAR

IF NOT __ISVALIDREF(fbIED) THEN
    RETURN;
END_IF

qValue := WORD_To_AcsiQuality(0);
sValue := AcsiQuality_To_String(qValue); (* sValue := 'Q#00|00000000|0|00' *)
nValue := AcsiQuality_To_WORD(qValue); (* nValue := 2#00000000_00000000 *)
IF fbIED.Relay.LLN0.Beh.q.nValue <> nValue THEN
    qValue := fbIED.Relay.LLN0.Beh.q.qValue;
END_IF

qValue := String_To_AcsiQuality('Q#01|00110011|0|01');
sValue := AcsiQuality_To_String(qValue); (* sValue := 'Q#01|00110011|0|01' *)
nValue := AcsiQuality_To_Word(qValue); (* nValue := 2#00010011_00110010 *)
IF fbIED.Relay.LLN0.Health.q.nValue <> nValue THEN
    qValue := fbIED.Relay.LLN0.Health.q.qValue;
END_IF

qValue := String_To_AcsiQuality('Q#10|11001100|1|10');
sValue := AcsiQuality_To_String(qValue); (* sValue := 'Q#10|11001100|1|10' *)
nValue := AcsiQuality_To_Word(qValue); (* nValue := 2#00001100_11001101 *)
IF fbIED.Relay.LLN0.Mod_.q.nValue <> nValue THEN
    qValue := fbIED.Relay.LLN0.Mod_.q.qValue;
END_IF

AcsiQuality_ToFrom_WORD := TRUE;
    
```

6.4.69 ZeroOctetString

Überschreibt alle Oktette eines Oktett-Strings mit dem Wert: 0.

Namensraum: [Tc3_Collections](#)  [96](#)

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

```

FUNCTION ZeroOctetString : UDINT
VAR_IN_OUT
    o : ARRAY[*] OF BYTE;
END_VAR
    
```

 Ein-/Ausgänge

| Name | Typ | Beschreibung |
|------|------------------|---|
| o | ARRAY[*] OF BYTE | Oktett-String dessen Oktette mit 0 überschrieben werden sollen. |

 Rückgabewert

| Name | Typ | Beschreibung |
|-----------------|-------|-------------------------------------|
| ZeroOctetString | UDINT | Anzahl der überschriebenen Oktetts. |

Beispiel

```

METHOD FINAL Sample_ZeroOctetString : BOOL
VAR
  o6  : T_OCTET6:=[6(16#FF)];
  o8  : T_OCTET8:=[8(16#FF)];
  o64 : T_OCTET64:=[64(16#FF)];
  size: UDINT;
END_VAR

size:=ZeroOctetString(o:=o6);(* size:=6, all octets: 0 *)
size:=ZeroOctetString(o:=o8);(* size:=8, all octets: 0 *)
size:=ZeroOctetString(o:=o64);(* size:=64, all octets: 0 *)
Sample_ZeroOctetString:=TRUE;

```

6.5 Schnittstellen

6.5.1 Datenmodell

6.5.1.1 Datenattribute

6.5.1.1.1 I_AcsiDATypeScaledValueConfig

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]**Bibliothek:** Tc3_Acsi (Tc3_Acsi.compiled-library)**Vererbungshierarchie**

[I_AcsiCommonNodeClass](#) [[▶ 320](#)] -> [I_AcsiCommonDataClass](#) [[▶ 311](#)] -> [I_AcsiCommonAttributeClass](#) [[▶ 310](#)] -> [I_AcsiDATypeScaledValueConfig](#)

```
INTERFACE I_AcsiDATypeScaledValueConfig EXTENDS I_AcsiCommonAttributeClass
```

Methoden

| Name | Beschreibung |
|---|--|
| ScaleAToP [▶ 308] | Skaliert einen Attributwert in einen Prozesswert |
| ScalePToA [▶ 309] | Skaliert einen Prozesswert in einen Attributwert |

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------------|------|---------|---------------------------------|
| fOffset | REAL | Get,Set | Attributwert: Skalierungsoffset |
| fScaleFactor | REAL | Get,Set | Attributwert: Skalierungsfaktor |

6.5.1.1.1.1 ScaleAToP

Skaliert einen Attributwert vom Typ DINT (Attribut i von AnalogueValue) in einen Prozesswert vom Typ LREAL. Die Konvertierungsformel lautet:

$$ScaleAToP := (iValue * scaleFactor) + offset$$

```

METHOD ScaleAToP : LREAL
VAR_INPUT
  iValue : DINT;
END_VAR

```

Rückgabeparameter: Skalierter Prozesswert

6.5.1.1.1.2 ScalePToA

Skaliert einen Prozesswert vom Typ LREAL in den Attributwert vom Typ DINT (Attribut i von AnalogueValue). Die Konvertierungsformel lautet:

$$ScalePToA := \frac{fProcess - offset}{scaleFactor}$$

```
METHOD ScalePToA : DINT
VAR_INPUT
    fProcess : LREAL;
END_VAR
```

Rückgabeparameter: Skalierter Attributwert

6.5.1.1.2 I_AcsiDATypeUnit

Namensraum: Tc3_Acsi [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[I_AcsiCommonNodeClass \[\[▶ 320\]\(#\)\]](#) -> [I_AcsiCommonDataClass \[\[▶ 311\]\(#\)\]](#) -> [I_AsciCommonAttributeClass \[\[▶ 310\]\(#\)\]](#) -> [I_AcsiDATypeUnit](#)

```
INTERFACE I_AcsiDATypeUnit EXTENDS I_AcsiCommonDataClass, I_AsciCommonAttributeClass
```

Methoden

| Name | Beschreibung |
|---|--|
| ScaleAToP [▶ 309] | Skaliert einen Attributwert in einen Prozesswert |
| ScalePToA [▶ 309] | Skaliert einen Prozesswert in einen Attributwert |

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|-------------|--|---------|-----------------------------|
| eMultiplier | E_AcsiMultiplier [▶ 416] | Get,Set | Attributwert: Multiplikator |
| eSIUnit | E_AcsiSIUnit [▶ 432] | Get,Set | Attributwert: SI-Einheit |

6.5.1.1.2.1 ScaleAToP

Skaliert einen Attributwert vom Typ REAL (Attribut f von AnalogueValue) in einen Prozesswert vom Typ LREAL. Die Konvertierungsformel lautet:

$$ScaleAToP := fValue * 10^{units.multiplier}$$

```
METHOD ScaleAToP : LREAL
VAR_INPUT
    fValue : REAL;
END_VAR
```

Rückgabeparameter: Skalierter Prozesswert

6.5.1.1.2.2 ScalePToA

Skaliert einen Prozesswert vom Typ LREAL in einen Attributwert vom Typ REAL (Attribut f von AnalogueValue). Die Konvertierungsformel lautet:

$$ScalePToA := \frac{fProcess}{10^{units.multiplier}}$$

```
METHOD ScalePToA : REAL
VAR_INPUT
    fProcess : LREAL;
END_VAR
```

Rückgabeparameter: Skalierter Attributwert

6.5.1.2 I_AcsiCommonAttributeClass

Objekte, die diese Schnittstelle implementieren befinden sich mindestens in der fünften Hierarchieebene des IEC 61850 Datenmodells und werden als Datenattribute (Data Attributes, DA) bezeichnet. Datenattribute bilden die kleinsten Objekteinheiten und beinhalten Detailinformationen wie Werte der Datenobjekte. Mehrere Datenattribute werden oft zu größeren Einheiten (Gemeinsamen Datenklassen, Common Data Classes, CDC) zusammengefasst (z. B. Binary Counter Reading, BCR). Es gibt auch Attribute, die wiederum selbst Attribute beinhalten (z. B. AnalogueValue, RangeConfig oder Unit). Solche Attribute werden als „Constructed Attributes“ bezeichnet. Dadurch können weitere, tiefere Hierarchieebenen entstehen. Der TwinCAT Telecontrol Configurator generiert für diese Ebenen Funktionsbausteine, die von [FB_AcsiCommonAttributeClass](#) [▶ 146] abgeleitet/erweitert werden. Alle Funktionsbausteine, die von [FB_AcsiCommonAttributeClass](#) [▶ 146] abgeleitet werden, besitzen automatisch eine Default-Implementierung der oben genannten Schnittstelle.

Auf Objekte, die diese Schnittstelle implementieren, können die gleichen Dienste wie bei der [I_AcsiCommonDataClass](#) [▶ 311]-Schnittstelle angewendet werden (z.B. GetDataValues, SetDataValues).

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[I_AcsiCommonNodeClass](#) [▶ 320] -> [I_AcsiCommonDataClass](#) [▶ 311] -> [I_AcsiCommonAttributeClass](#)

```
INTERFACE I_AcsiCommonAttributeClass EXTENDS I_AcsiCommonDataClass
```

Methoden

| Name | Beschreibung |
|---|---|
| AddAttributeToContainer [▶ 310] | Fügt ein Datenattributobjekt dem Container eines anderen Datenattributobjekts hinzu |

6.5.1.2.1 AddAttributeToContainer

Mit dieser Methode wird ein Datenattributobjekt (DA) dem Container eines anderen Datenattributobjekts (DA) hinzugefügt.

```
METHOD AddAttributeToContainer : BOOL
VAR_INPUT
    ipAttribute : I_AcsiCommonAttributeClass;
END_VAR
```

ipAttribute: Schnittstellenzeiger vom Typ [I_AcsiCommonAttributeClass](#) [▶ 310] des Datenattributobjekts, welches dem Container hinzugefügt werden soll.

Rückgabeparameter: Positive Meldung (TRUE) bei Erfolg und negative Meldung (FALSE) bei einem Fehler.

6.5.1.3 I_AcsiCommonBufferedReportControlBlockClass

Objekte, die diese Schnittstelle implementieren, sind Berichtssteuerblockinstanzen für gepufferte Reports (Buffered Report Control Blocks, BRCBs). Der TwinCAT Telecontrol Configurator generiert Berichtsteuerblockinstanzen, die von [FB_AcsiCommonBufferedReportControlBlockClass](#) [▶ 147] abgeleitet/erweitert werden. Alle Funktionsbausteine, die von [FB_AcsiCommonBufferedReportControlBlockClass](#) [▶ 147] abgeleitet werden, besitzen automatisch eine Default-Implementierung der oben genannten Schnittstelle.

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[I_AcsiCommonNodeClass \[▶ 320\]](#) -> [I_AcsiCommonDataClass \[▶ 311\]](#) -> [I_AcsiCommonControlBlockClass \[▶ 311\]](#) -> [I_AcsiCommonBufferedReportControlBlockClass](#)

```
INTERFACE I_AcsiCommonBufferedReportControlBlockClass EXTENDS I_AcsiCommonControlBlockClass
```

6.5.1.4 I_AcsiCommonControlBlockClass

Alle Objekte, bei denen es sich um Steuerblockinstanzen handelt, implementieren eine Schnittstelle, die von dieser Schnittstelle erbt. Diese Schnittstelle dient als Basisschnittstelle für alle Steuerblöcke.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[I_AcsiCommonNodeClass \[▶ 320\]](#) -> [I_AcsiCommonDataClass \[▶ 311\]](#) -> [I_AcsiCommonControlBlockClass](#)

```
INTERFACE I_AcsiCommonControlBlockClass EXTENDS I_AcsiCommonDataClass
```

6.5.1.5 I_AcsiCommonControllableDataClass

Auf Objekten, die diese Schnittstelle implementieren, können Dienste für die Steuerung wie Control, Operate, Cancel, Select oder SelectWithValue ausgeführt werden. Der TwinCAT Telecontrol Configurator generiert hierfür Funktionsbausteine, die von [FB_AcsiCommonControllableDataClass \[▶ 148\]](#) abgeleitet/erweitert werden. Alle Funktionsbausteine, die von [FB_AcsiCommonControllableDataClass \[▶ 148\]](#) abgeleitet werden, besitzen automatisch eine Default-Implementierung der oben genannten Schnittstelle.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[I_AcsiCommonNodeClass \[▶ 320\]](#) -> [I_AcsiCommonDataClass \[▶ 311\]](#) -> [I_AcsiCommonControllableDataClass](#)

```
INTERFACE I_AcsiCommonControllableDataClass EXTENDS I_AcsiCommonDataClass
```

6.5.1.6 I_AcsiCommonDataClass

Objekte, die diese Schnittstelle implementieren, befinden sich in der vierten Hierarchieebene des IEC 61850 Datenmodells und werden als Datenobjekte (Data Objects, DO) bezeichnet. IEC 61850-7-4 definiert den Umfang der Datenobjekte. Einige Datenobjekte müssen zwingend enthalten sein (Mandatory), andere wiederum sind optional. Datenobjekte können auch als Container für andere Datenobjekte dienen. Der TwinCAT Telecontrol Configurator generiert für diese Ebenen Funktionsbausteine, die von [FB_AcsiCommonDataClass \[▶ 148\]](#) abgeleitet/erweitert werden. Alle Funktionsbausteine, die von [FB_AcsiCommonDataClass \[▶ 148\]](#) abgeleitet werden, besitzen automatisch eine Default-Implementierung der oben genannten Schnittstelle. Auf Objekte, die diese Schnittstelle implementieren, können Dienste wie z. B. GetDataValues, SetDataValues angewendet werden.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[I_AcsiCommonNodeClass \[▶ 320\]](#) -> [I_AcsiCommonDataClass](#)

```
INTERFACE I_AcsiCommonDataClass EXTENDS I_AcsiCommonNodeClass
```

Methoden

| Name | Beschreibung |
|---|---|
| AddAttributeToContainer [▶ 312] | Fügt ein Datenattributobjekt dem Container eines Datenobjekts hinzu |
| AddDataToContainer [▶ 312] | Fügt ein Datenobjekt dem Container eines anderen Datenobjekts hinzu |
| AddToDataSet [▶ 312] | Fügt ein Datenobjekt dem Membercontainer des Datensatzes hinzu |
| DeleteFromDataSet [▶ 313] | Entfernt ein Datenobjekt aus dem Membercontainer des Datensatzes |

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|---|-------|---------|------------------------|
| bPresence [▶ 313] | BOOL | Get,Set | Attribut-Sichtbarkeit |
| Fc [▶ 313] | DWORD | Get,Set | Funktionale Gruppe |
| TrgOp [▶ 313] | DWORD | Get,Set | Auslöser eines Reports |

6.5.1.6.1 AddAttributeToContainer

Mit dieser Methode wird ein Datenattributobjekt (DA) dem Container eines Datenobjekts (DO) hinzugefügt.

```
METHOD AddAttributeToContainer : BOOL
VAR_INPUT
    ipAttribute : I_AcsiCommonAttributeClass;
END_VAR
```

ipAttribute: Schnittstellenzeiger vom Typ [I_AcsiCommonAttributeClass](#) [[▶ 310](#)] des Datenattributobjekts, welches dem Container hinzugefügt werden soll.

Rückgabeparameter: Positive Meldung (TRUE) bei Erfolg und negative Meldung (FALSE) bei einem Fehler.

6.5.1.6.2 AddDataToContainer

Mit dieser Methode wird ein Datenobjekt (DO) dem Container eines anderen Datenobjekts (DO) hinzugefügt.

```
METHOD FINAL AddDataToContainer : BOOL
VAR_INPUT
    ipData : I_AcsiCommonDataClass;
END_VAR
```

ipData: Schnittstellenzeiger vom Typ [I_AcsiCommonDataClass](#) [[▶ 311](#)] des Datenobjekts welches dem Container hinzugefügt werden soll.

Rückgabeparameter: Positive Meldung (TRUE) bei Erfolg und negative Meldung (FALSE) bei einem Fehler.

6.5.1.6.3 AddToDataSet

Mit dieser Methode wird ein Datenobjekt (DO) dem Membercontainer des Datensatzes (Data-Set) hinzugefügt.

```
METHOD FINAL AddToDataSet : BOOL
VAR_INPUT
    ipDataSet : I_AcsiCommonDataSetClass;
    eFc       : E_AcsiFc;
END_VAR
VAR_OUTPUT
    bMatched : BOOL;
    ipDirNode : I_AcsiDataSetMemberClass;
END_VAR
```

ipDataSet: Schnittstellenzeiger vom Typ [I_AcsiCommonDataSetClass](#) [[▶ 313](#)] des Datensatzes, in dessen Membercontainer das Datenobjekt hinzugefügt werden soll.

eFc: Funktionale Gruppe des Datenobjekts vom Typ [E_AcsiFc](#) [[▶ 409](#)].

bMatched: Bei TRUE ist das Datenobjekt bereits im Membercontainer vorhanden.

ipDirNode: Schnittstellenzeiger des neuen Container-Listenelements.

Rückgabeparameter: Positive Meldung (TRUE) bei Erfolg und negative Meldung (FALSE) bei einem Fehler.

6.5.1.6.4 DeleteFromDataSet

Mit dieser Methode wird ein Datenobjekt (DO) aus dem Membercontainer des Datensatzes (Data-Set) entfernt.

```
METHOD FINAL DeleteFromDataSet : BOOL
VAR_INPUT
    ipDataSet : I_AcsiCommonDataSetClass;
    eFc       : E_AcsiFc;
END_VAR
```

ipDataSet: Schnittstellenzeiger vom Typ [I_AcsiCommonDataSetClass](#) [[▶ 313](#)] des Datensatzes, aus dessen Membercontainer das Datenobjekt entfernt werden soll.

eFc: Funktionale Gruppe des Datenobjekts vom Typ [E_AcsiFc](#) [[▶ 409](#)].

Rückgabeparameter: Positive Meldung (TRUE) bei Erfolg und negative Meldung (FALSE) bei einem Fehler.

6.5.1.6.5 bPresence

Diese Eigenschaft bestimmt, ob die Instanz eines Datenobjekts oder Datenattributes zwingend notwendig oder optional ist. Beim Setzen der Sichtbarkeit eines verschachtelten (constructed) Datenattributs wird auch die Sichtbarkeit der unterlagerten Datenattribute auf den gleichen Wert gesetzt.

```
PROPERTY bPresence : BOOL {GET,SET}
```

bPresence: Sichtbarkeit/Existenz:

- TRUE := Mandatory;
- FALSE := Optional;

6.5.1.6.6 TrgOp

Diese Eigenschaft bestimmt den Auslöser eines Reports (spontane Übertragung der Daten eines Datenobjekts oder Datenattributes).

```
PROPERTY TrgOp : DWORD {GET,SET}
```

TrgOp: Auslöser eines Reports. Ein oder mehrere Werte können „verodert“ werden.

6.5.1.6.7 Fc

Diese Eigenschaft bestimmt, zu welcher funktionalen Gruppe ein Datenobjekt oder Datenattribut gehört. Beim Setzen der funktionalen Gruppe eines verschachtelten (constructed) Datenattributs wird auch die Sichtbarkeit der unterlagerten Datenattribute auf den gleichen Wert gesetzt.

```
PROPERTY Fc : DWORD {GET,SET}
```

Fc: Funktionale Gruppe. Ein oder mehrere Werte können „verodert“ werden.

6.5.1.7 I_AcsiCommonDataSetClass

Objekte, die diese Schnittstelle implementieren, sind Datensatzinstanzen (Data-Sets). TwinCAT Telecontrol Configurator generiert Datensatzinstanzen, die von [FB_AcsiCommonDataSetClass](#) [[▶ 149](#)] abgeleitet/erweitert werden. Alle Funktionsbausteine, die von [FB_AcsiCommonDataSetClass](#) [[▶ 149](#)] abgeleitet werden, besitzen automatisch eine Default-Implementierung der oben genannten Schnittstelle.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: [Tc3_Acsi](#) (Tc3_Acsi.compiled-library)

Vererbungshierarchie

I_AcsiCommonNodeClass [▶ 320] -> I_AcsiCommonDataSetClass

```
INTERFACE I_AcsiCommonDataSetClass EXTENDS I_AcsiCommonNodeClass
```

Methoden

| Name | Beschreibung |
|-----------------------------|--|
| AddMember [▶ 314] | Fügt ein neues Datenobjekt dem Membercontainer hinzu |
| DeleteMember [▶ 314] | Entfernt ein Datenobjekt aus dem Membercontainer |
| DeleteAllMembers
[▶ 315] | Entfernt alle Datenobjekte aus dem Membercontainer |

Methoden zur Ereignisbehandlung

| Name | Beschreibung |
|----------------|--|
| OnInit [▶ 315] | Initialisiert/Konfiguriert die Datenobjekte im Membercontainer |

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|----------|-------|---------|--|
| nMembers | UDINT | Get | Anzahl der Datenobjekte im Membercontainer |

6.5.1.7.1 AddMember

Mit dieser Methode wird ein Datenobjekt (DO) dem Membercontainer eines Datensatzes (Data-Set) hinzugefügt.

```
METHOD FINAL AddMember : BOOL
VAR_INPUT
    ipData    : I_AcsiCommonDataClass;
    eFc       : E_AcsiFc;
END_VAR
VAR_OUTPUT
    bMatched : BOOL;
    ipDirNode: I_AcsiDataSetMemberClass;
END_VAR
```

ipData: Schnittstellenzeiger vom Typ [I_AcsiCommonDataClass \[▶ 311\]](#) des Datenobjekts, welches dem Membercontainer hinzugefügt werden soll.

eFc: Funktionale Gruppe des Datenobjekts vom Typ [E_AcsiFc \[▶ 409\]](#).

bMatched: Bei TRUE ist das Datenobjekt bereits im Membercontainer vorhanden.

ipDirNode: Schnittstellenzeiger des neuen Container-Listenelements.

Rückgabeparameter: Positive Meldung (TRUE) bei Erfolg und negative Meldung (FALSE) bei einem Fehler.

6.5.1.7.2 DeleteMember

Mit dieser Methode wird ein Datenobjekt (DO) aus dem Membercontainer des Datensatzes (Data-Set) entfernt.

```
METHOD FINAL DeleteMember : BOOL
VAR_INPUT
    ipData    : I_AcsiCommonDataClass;
    eFc       : E_AcsiFc;
END_VAR
```

ipData: Schnittstellenzeiger vom Typ [I_AcsiCommonDataClass \[▶ 311\]](#) des Datenobjekts, welches aus dem Membercontainer entfernt werden soll.

eFc: Funktionale Gruppe des Datenobjekts vom Typ [E_AcsiFc](#) [[▶ 409](#)].

Rückgabeparameter: Positive Meldung (TRUE) bei Erfolg und negative Meldung (FALSE) bei einem Fehler.

6.5.1.7.3 DeleteAllMembers

Mit dieser Methode werden alle Datenobjekte (DO) und Datenattribute (DA) aus dem Membercontainer des Datensatzes (Data-Set) entfernt.

```
METHOD FINAL DeleteAllMembers : BOOL
VAR_OUTPUT
    nDeleted : UDINT;
END_VAR
```

nDeleted: Anzahl der gelöschten Datenobjekt- und Datenattribute Member.

Rückgabeparameter: Positive Meldung (TRUE) bei Erfolg und negative Meldung (FALSE) bei einem Fehler.

6.5.1.7.4 OnInit

Diese Methode kann, muss aber nicht, von der Applikation überladen werden. Sie kann dafür verwendet werden, um den Membercontainer des Datensatzes mit Datenobjekt- und Datenattributereferenzen zu füllen.

```
METHOD OnInit : BOOL
```

Rückgabeparameter: Positive Meldung (TRUE) bei Erfolg und negative Meldung (FALSE) bei einem Fehler.

6.5.1.8 I_AcsiCommonGooseControlBlockClass

Objekte, die diese Schnittstelle implementieren, sind Steuerblockinstanzen für Generic Object Oriented Substation Events (GOOSE). Der TwinCAT Telecontrol Configurator generiert GOOSE-Steuerblockinstanzen, die von [FB_AcsiCommonGooseControlBlockClass](#) [[▶ 150](#)] abgeleitet/erweitert werden. Alle Funktionsbausteine, die von [FB_AcsiCommonGooseControlBlockClass](#) [[▶ 150](#)] abgeleitet werden, besitzen automatisch eine Default-Implementierung der oben genannten Schnittstelle.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[I_AcsiCommonNodeClass](#) [[▶ 320](#)] -> [I_AcsiCommonDataClass](#) [[▶ 311](#)] -> [I_AcsiCommonControlBlockClass](#) [[▶ 311](#)] -> [I_AcsiCommonGooseControlBlockClass](#)

```
INTERFACE I_AcsiCommonGooseControlBlockClass EXTENDS I_AcsiCommonControlBlockClass
```

6.5.1.9 I_AcsiCommonIEDGroupClass

Objekte, die diese Schnittstelle implementieren repräsentieren eine Gruppe von mehreren IEDs. Alle IEDs in dieser Gruppe verwenden für die Kommunikation den gleichen „Real-Time Ethernet Adapter (Multiple Protocol Handler)“. Diese Objekte dienen als Container für mehrere Server-Instanzen (z.B. mehrere GOOSE Subscriber oder Publisher).

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[__System.IQueryInterface](#) -> [I_AcsiCommonIEDGroupClass](#)

```
INTERFACE I_AcsiCommonIEDGroupClass EXTENDS __System.IQueryInterface
```

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|-------------|--|---------|--------------------------------|
| nIEDs | UDINT | Get | Anzahl der IEDs im Container. |
| sObjectName | T_AcsiObjectName [▶ 475] | Get | Objektname. |
| Tag | T_AcsiTag [▶ 476] | Get | Zusätzliche Objektinformation. |

6.5.1.10 I_AcsiCommonIntelligentElectronicDeviceClass

Objekte, die diese Schnittstelle implementieren, befinden sich in der obersten Hierarchieebene des IEC 61850 Datenmodells. Diese Ebene kann auch als Server-Instanz bezeichnet werden. Der TwinCAT Telecontrol Configurator generiert für diese Ebene Funktionsbausteine, die von [FB_AcsiCommonIntelligentElectronicDeviceClass \[▶ 151\]](#) abgeleitet/erweitert werden. Alle Funktionsbausteine die von [FB_AcsiCommonIntelligentElectronicDeviceClass \[▶ 151\]](#) abgeleitet werden, besitzen automatisch eine Default-Implementierung der oben genannten Schnittstelle.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[I_AcsiCommonNodeClass \[▶ 320\]](#) -> [I_AcsiCommonIntelligentElectronicDeviceClass](#)

```
INTERFACE I_AcsiCommonIntelligentElectronicDeviceClass EXTENDS I_AcsiCommonNodeClass
```

 **Methoden**

| Name | Beschreibung |
|---|--|
| AddLogicalDeviceToContainer [▶ 316] | Fügt ein neues Logisches Gerät dem Container des Servers hinzu |

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|-----------------|-------|---------|--|
| nLogicalDevices | UDINT | Get | Anzahl der Logischen Geräte im Container des Servers |

6.5.1.10.1 AddLogicalDeviceToContainer

Mit dieser Methode wird eine Objektinstanz des Logischen Geräts (Logical Device, LD) dem Container des Servers (Intelligent Logical Device, IED) hinzugefügt.

```
METHOD FINAL AddLogicalDeviceToContainer : BOOL
VAR_INPUT
    ipLogicalDevice : I_AcsiCommonLogicalDeviceClass;
END_VAR
VAR_OUTPUT
    bMatched       : BOOL;
    ipDirNode      : I_AcsiLogicalDeviceContainerClass;
END_VAR
```

ipLogicalDevice: Schnittstellenzeiger vom Typ [I_AcsiCommonLogicalDeviceClass \[▶ 317\]](#) des Logischen Geräts, welches dem Container hinzugefügt werden soll.

bMatched: Bei TRUE existiert das Objekt bereits im Container.

ipDirNode: Schnittstellenzeiger des neuen Container-Listenelements.

Rückgabeparameter: Positive Meldung (TRUE) bei Erfolg und negative Meldung (FALSE) bei einem Fehler.

6.5.1.11 I_AcsiCommonLogicalDeviceClass

Objekte, die diese Schnittstelle implementieren, befinden sich in der zweiten Hierarchieebene des IEC 61850 Datenmodells. Diese Ebene wird als Logisches Gerät (Logical Device, LD) bezeichnet. Der TwinCAT Telecontrol Configurator generiert für diese Ebene Funktionsbausteine, die von [FB_AcsiCommonLogicalDeviceClass \[▶ 152\]](#) abgeleitet/erweitert werden. Alle Funktionsbausteine, die von [FB_AcsiCommonLogicalDeviceClass \[▶ 152\]](#) abgeleitet werden besitzen eine Default-Implementierung der oben genannten Schnittstelle.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[I_AcsiCommonNodeClass \[▶ 320\]](#) -> [I_AcsiCommonLogicalNodeClass](#)

```
INTERFACE I_AcsiCommonLogicalDeviceClass EXTENDS I_AcsiCommonNodeClass
```

Methoden

| Name | Beschreibung |
|---|---|
| AddLogicalNodeToContainer [▶ 317] | Fügt einen neuen Logischen Knoten hinzu |

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|---------------|-------|---------|--|
| nLogicalNodes | UDINT | Get | Anzahl der Logischen Knoten im Container des Logischen Gerätes |

6.5.1.11.1 AddLogicalNodeToContainer

Mit dieser Methode wird eine Objektinstanz des Logischen Knoten (Logical Node, LN) dem Container des Logischen Gerätes (Logical Device, LD) hinzugefügt.

```
METHOD FINAL AddLogicalNodeToContainer : BOOL
VAR_INPUT
    ipLogicalNode : I_AcsiCommonLogicalNodeClass;
END_VAR
VAR_OUTPUT
    bMatched      : BOOL;
    ipDirNode     : I_AcsiLogicalNodeContainerClass;
END_VAR
```

ipLogicalNode: Schnittstellenzeiger vom Typ [I_AcsiCommonLogicalNodeClass \[▶ 317\]](#) des Logischen Knoten, welcher dem Container hinzugefügt werden soll.

bMatched: Bei TRUE existiert das Objekt bereits im Container.

ipDirNode: Schnittstellenzeiger des neuen Container-Listenelements.

Rückgabeparameter: Positive Meldung (TRUE) bei Erfolg und negative Meldung (FALSE) bei einem Fehler.

6.5.1.12 I_AcsiCommonLogicalNodeClass

Objekte, die diese Schnittstelle implementieren, befinden sich in der dritten Hierarchieebene des IEC 61850 Datenmodells. Diese Ebene wird als Logischer Knoten (Logical Node, LN) bezeichnet. Der TwinCAT Telecontrol Configurator generiert für diese Ebene Funktionsbausteine, die von [FB_AcsiCommonLogicalNodeClass \[▶ 153\]](#) abgeleitet/erweitert werden. Alle Funktionsbausteine, die von [FB_AcsiCommonLogicalNodeClass \[▶ 153\]](#) abgeleitet werden, besitzen eine Default-Implementierung der oben genannten Schnittstelle.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

I_AcsiCommonNodeClass [▶ 320] -> I_AcsiCommonLogicalNodeClass

INTERFACE I_AcsiCommonLogicalNodeClass EXTENDS I_AcsiCommonNodeClass

Methoden

| Name | Beschreibung |
|--|--|
| AddDataToContainer [▶ 318] | Fügt ein neues Datenobjekt hinzu |
| AddDataSetToContainer [▶ 318] | Fügt einen neuen Datensatz hinzu |
| AddUnbufferedReportControlBlockToContainer [▶ 319] | Fügt eine neue Berichtssteuerblockinstanz für nicht gepufferte Reports hinzu |
| AddBufferedReportControlBlockToContainer [▶ 319] | Fügt eine neue Berichtssteuerblockinstanz für gepufferte Reports hinzu |
| AddGooseControlBlockToContainer [▶ 319] | Fügt eine neue Goose-Steuerblockinstanz hinzu |

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------------------------------|-------|---------|--|
| nDataSets | UDINT | Get | Anzahl der Datensätze im Container des Logischen Knoten |
| nBufferedReportControlBlocks | UDINT | Get | Anzahl der Berichtssteuerblockinstanzen für nicht gepufferte Reports |
| nUnbufferedReportControlBlocks | UDINT | Get | Anzahl der Berichtssteuerblockinstanzen für gepufferte Reports |
| nGooseControlBlocks | UDINT | Get | Anzahl der Goose-Steuerblockinstanzen |

6.5.1.12.1 AddDataToContainer

Mit dieser Methode wird ein Datenobjekt (DO) dem Container des Logischen Knoten (Logical Node, LN) hinzugefügt.

```
METHOD FINAL AddDataToContainer : BOOL
VAR_INPUT
    ipData      : I_AcsiCommonDataClass;
END_VAR
VAR_OUTPUT
    bMatched    : BOOL;
    ipDirNode   : I_AcsiDataContainerClass;
END_VAR
```

ipData: Schnittstellenzeiger vom Typ [I_AcsiCommonDataClass \[▶ 311\]](#) des Datenobjekts welches dem Container hinzugefügt werden soll.

bMatched: Bei TRUE existiert das Objekt bereits im Container.

ipDirNode: Schnittstellenzeiger des neuen Container-Listenelements.

Rückgabeparameter: Positive Meldung (TRUE) bei Erfolg und negative Meldung (FALSE) bei einem Fehler.

6.5.1.12.2 AddDataSetToContainer

Mit dieser Methode wird eine Objektinstanz des Datensatzes (Data-Set) dem Container des Logischen Knoten (Logical Node, LN) hinzugefügt.

```
METHOD FINAL AddDataSetToContainer : BOOL
VAR_INPUT
    ipDataSet   : I_AcsiCommonDataSetClass;
END_VAR
VAR_OUTPUT
```

```

    bMatched : BOOL;
    ipDirNode : I_AcsiDataSetContainerClass;
END_VAR

```

ipDataSet: Schnittstellenzeiger vom Typ [I_AcsiCommonDataSetClass](#) [[▶ 313](#)] des Datensatzes, welcher dem Container hinzugefügt werden soll.

bMatched: Bei TRUE existiert das Objekt bereits im Container.

ipDirNode: Schnittstellenzeiger des neuen Container-Listenelements.

Rückgabeparameter: Positive Meldung (TRUE) bei Erfolg und negative Meldung (FALSE) bei einem Fehler.

6.5.1.12.3 AddUnbufferedReportControlBlockToContainer

Mit dieser Methode wird eine Berichtssteuerblockinstanz für nicht gepufferte Reports (Unbuffered Report Control Block, URCB) dem Container des Logischen Knoten (Logical Node, LN) hinzugefügt.

```

METHOD FINAL AddUnbufferedReportControlBlockToContainer : BOOL
VAR_INPUT
    ipUnbufferedReportControlBlock : I_AcsiCommonUnbufferedReportControlBlockClass;
END_VAR
VAR_OUTPUT
    bMatched           : BOOL;
    ipDirNode          : I_AcsiUrCBContainerClass;
END_VAR

```

ipUnbufferedReportControlBlock: Schnittstellenzeiger vom Typ [I_AcsiUnbufferedReportControlBlockClass](#) [[▶ 321](#)] der Berichtssteuerblockinstanz, welche dem Container hinzugefügt werden soll.

bMatched: Bei TRUE existiert das Objekt bereits im Container.

ipDirNode: Schnittstellenzeiger des neuen Container-Listenelements.

Rückgabeparameter: Positive Meldung (TRUE) bei Erfolg und negative Meldung (FALSE) bei einem Fehler.

6.5.1.12.4 AddBufferedReportControlBlockToContainer

Mit dieser Methode wird eine Berichtssteuerblockinstanz für gepufferte Reports (Buffered Report Control Block, BRCB) dem Container des Logischen Knoten (Logical Node, LN) hinzugefügt.

```

METHOD FINAL AddBufferedReportControlBlockToContainer : BOOL
VAR_INPUT
    ipBufferedReportControlBlock : I_AcsiCommonBufferedReportControlBlockClass;
END_VAR
VAR_OUTPUT
    bMatched           : BOOL;
    ipDirNode          : I_AcsiBrCBContainerClass;
END_VAR

```

ipBufferedReportControlBlock: Schnittstellenzeiger vom Typ [I_AcsiBufferedReportControlBlockClass](#) [[▶ 310](#)] der Berichtssteuerblockinstanz welche dem Container hinzugefügt werden soll.

bMatched: Bei TRUE existiert das Objekt bereits im Container.

ipDirNode: Schnittstellenzeiger des neuen Container-Listenelements.

Rückgabeparameter: Positive Meldung (TRUE) bei Erfolg und negative Meldung (FALSE) bei einem Fehler.

6.5.1.12.5 AddGooseControlBlockToContainer

Mit dieser Methode wird eine Goose-Steuerblockinstanz (Generic Object Oriented Substation Event Control Block, GCB) dem Container des Logischen Knoten (Logical Node, LN) hinzugefügt.

```

METHOD FINAL AddGooseControlBlockToContainer : BOOL
VAR_INPUT
    ipGooseControlBlock : I_AcsiCommonGooseControlBlockClass;
END_VAR
VAR_OUTPUT

```

```

    bMatched          : BOOL;
    ipDirNode         : I_AcsiGoCBContainerClass;
END_VAR

```

ipGooseControlBlock: Schnittstellenzeiger vom Typ `I_AcsiCommonGooseControlBlockClass` der Goose-Steuerblockinstanz welche dem Container hinzugefügt werden soll.

bMatched: Bei TRUE existiert das Objekt bereits im Container.

ipDirNode: Schnittstellenzeiger des neuen Container-Listenelements.

Rückgabeparameter: Positive Meldung (TRUE) bei Erfolg und negative Meldung (FALSE) bei einem Fehler.

6.5.1.13 I_AcsiCommonNodeClass

Objekte, die diese Schnittstelle implementieren, bilden die Grundlage des TwinCAT IEC 61850 Datenmodells. Alle Funktionsbausteine, die TwinCAT Telecontrol Configurator generiert, implementieren diese Schnittstelle oder werden von dem Funktionsbaustein `FB_AcsiCommonNodeClass` [▶ 154] abgeleitet, der diese Schnittstelle implementiert.

Namensraum: `Tc3_Acsi` [▶ 96]

Bibliothek: `Tc3_Acsi` (`Tc3_Acsi.compiled-library`)

Vererbungshierarchie

`I_AcsiCommonNodeClass`

```
INTERFACE I_AcsiCommonNodeClass
```

Methoden

| Name | Beschreibung |
|--|--|
| <code>GetObjectReference</code>
[▶ 320] | Liefert den Referenzpfad einer Objektinstanz |
| <code>GetServerObject</code> [▶ 321] | Liefert den Schnittstellenzeiger der Objektinstanz des Servers |

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|-------------------------------------|--|---------|--------------|
| <code>sObjectName</code>
[▶ 321] | <code>T_AcsiObjectName</code>
[▶ 475] | Get,Set | Objektname |

6.5.1.13.1 GetObjectReference

Mit dieser Methode kann der Referenzpfad einer Objektinstanz des TwinCAT IEC 61850 Datenmodells ermittelt werden. Die Abfrage der Objektinstanz des Servers (IEDs) liefert den Namen des Servers. Bei allen anderen Objektinstanzen oberhalb des Servers ist der Name des Servers in dem Referenzpfad nicht enthalten. Der Name der funktionalen Gruppe ist in dem Referenzpfad der Objektinstanzen oberhalb der Logischen Knoten (LNs) auch nicht enthalten.

```
METHOD GetObjectReference : T_AcsiObjectReference
```

Rückgabeparameter: String mit dem Referenzpfad der Objektinstanz.

Beispiel:

Auszug aus einem fiktiven TwinCAT IEC 61850 Datenmodell:

```

MA2C03          { IED }
  -MA2C03Relay  { LD }
    -LLN0       { LN }
      -Mod       { DO }
        stVal    { DA }
          q       { DA }

```

```

        t          {DA}
    +Beh          {DO}
    +Health       {DO}
    +NamPlt       {DO}
    +LPHD1        {LN}
    ...

```

Deklarationsteil:

```

PROGRAM MAIN
VAR
    sRef : T_AcsiObjectReference;
END_VAR

```

Implementierung:

```

sRef := MA2C03.GetObjectReference();           (**MA2C03**)
sRef := MA2C03.MA2C03Relay.GetObjectReference(); (**MA2C03Relay**)
sRef := MA2C03.MA2C03Relay.LLN0.GetObjectReference(); (**MA2C03Relay/LLN0**)
sRef := MA2C03.MA2C03Relay.LLN0.Mod_.GetObjectReference(); (**MA2C03Relay/LLN0.Mod**)
sRef := MA2C03.MA2C03Relay.LLN0.Mod_.stVal.GetObjectReference(); (**MA2C03Relay/LLN0.Mod.stVal**)
sRef := MA2C03.MA2C03Relay.LLN0.Mod_.q.GetObjectReference(); (**MA2C03Relay/LLN0.Mod.q**)
sRef := MA2C03.MA2C03Relay.LLN0.Mod_.t.GetObjectReference(); (**MA2C03Relay/LLN0.Mod.t**)
sRef := MA2C03.MA2C03Relay.LLN0.Beh.GetObjectReference(); (**MA2C03Relay/LLN0.Beh**)
sRef := MA2C03.MA2C03Relay.LLN0.Health.GetObjectReference(); (**MA2C03Relay/LLN0.Health**)
sRef := MA2C03.MA2C03Relay.LLN0.NamPlt.GetObjectReference(); (**MA2C03Relay/LLN0.NamPlt**)
sRef := MA2C03.MA2C03Relay.LPHD1.GetObjectReference(); (**MA2C03Relay/LPHD1**)

```

6.5.1.13.2 GetServerObject

Mit dieser Methode kann der Schnittstellenzeiger der Objektinstanz des Servers (oberste Ebene des IEC 61850 Datenmodells, IED) ermittelt werden.

```

METHOD GetServerObject : BOOL
VAR_OUTPUT
    ipServer : I_AcsiCommonIntelligentElectronicDeviceClass;
END_VAR

```

ipServer: Schnittstellenzeiger vom Typ [I_AcsiCommonIntelligentElectronicDeviceClass](#) [► 316] der Objektinstanz des Servers.

Rückgabeparameter: Positive Meldung (TRUE) bei Erfolg und negative Meldung (FALSE) bei einem Fehler.

6.5.1.13.3 sObjectName

```

PROPERTY sObjectName : T_AcsiObjectName (GET)

```

sObjectName: Objektname als String des Typs [T_AcsiObjectName](#) [► 475].

6.5.1.14 I_AcsiCommonUnbufferedReportControlBlockClass

Objekte, die diese Schnittstelle implementieren, sind Berichtssteuerblockinstanzen für nicht gepufferte Reports (Unbuffered Report Control Blocks, URCBs). Der TwinCAT Telecontrol Configurator generiert Berichtssteuerblockinstanzen, die von [FB_AcsiCommonUnbufferedReportControlBlockClass](#) [► 154] abgeleitet/erweitert werden. Alle Funktionsbausteine, die von [FB_AcsiCommonUnbufferedReportControlBlockClass](#) [► 154] abgeleitet werden, besitzen automatisch eine Default-Implementierung der oben genannten Schnittstelle.

Namensraum: [Tc3_Acsi](#) [► 96]

Bibliothek: [Tc3_Acsi](#) ([Tc3_Acsi.compiled-library](#))

Vererbungshierarchie

[I_AcsiCommonNodeClass](#) [► 320] -> [I_AcsiCommonDataClass](#) [► 311] -> [I_AcsiCommonControlBlockClass](#) [► 311] -> [I_AcsiCommonUnbufferedReportControlBlockClass](#)

```

INTERFACE I_AcsiCommonUnbufferedReportControlBlockClass EXTENDS I_AcsiCommonControlBlockClass

```

6.5.1.15 I_AcsiDATypeLastApplError

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Vererbungshierarchie

[I_AcsiCommonNodeClass](#) [[▶ 320](#)] -> [I_AcsiCommonDataClass](#) [[▶ 311](#)] -> [I_AcsiCommonAttributeClass](#) [[▶ 310](#)] -> [I_AcsiDATypeLastApplError](#)

```
INTERFACE I_AcsiDATypeLastApplError EXTENDS I_AcsiCommonAttributeClass
```



Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|-----------|--|---------|--------------|
| sCntrlObj | STRING(129) | Get,Set | Attributwert |
| eError | E_AcsiApplError [▶ 399] | Get,Set | Attributwert |
| stOrigin | ST_AcsiOriginator [▶ 460] | Get,Set | Attributwert |
| nCtlNum | BYTE | Get,Set | Attributwert |
| eAddCause | E_AcsiApplAddCause [▶ 398] | Get,Set | Attributwert |

6.5.2 Protokolleinstellungen

6.5.2.1 I_GseAdapterSettingsClass

Objekte, die diese Schnittstelle implementieren, dienen der Konfiguration eines für die GSE-Kommunikation nutzbaren Netzwerkadapters.

Namensraum: [Tc3_Gse](#) [[▶ 97](#)]

Bibliothek: Tc3_Gse (Tc3_Gse.compiled-library)

Vererbungshierarchie

[I_GseAdapterSettingsClass](#)

```
INTERFACE I_GseAdapterSettingsClass
```

 **Eigenschaften**

| Name | Typ | Zu-griff | Beschreibung |
|----------------|--------------------------------------|------------|---|
| bExecGoCBPubs | BOOL | Get | Aktiviert/Deaktiviert das automatische Aufrufen der „Execute“-Methode der Publisher GOOSE-Kontrollblöcke durch den Adapter. |
| bExecGoCBSubs | BOOL | Get | Aktiviert/Deaktiviert das automatische Aufrufen der „Execute“-Methode der Subscriber GOOSE-Kontrollblöcke durch den Adapter. |
| bSimulation | BOOL | Get | Aktiviert/Deaktiviert das GOOSE-Simulations-Bit Feld. |
| eAdapterPrio | TcEthernetAdapterPriority
[▶ 453] | Get | Priorität des Netzwerkadapters. |
| eDispatchMode | E_GseDispatchMode
[▶ 445] | Get | Einstellung, ob empfangene Ethernet-Telegramme an das Betriebssystem weitergeleitet werden. |
| macAddr | ETHERNET_ADDRESS
[▶ 451] | Get
Set | MAC-Adresse des Netzwerkadapters. |
| multicastAddr | ETHERNET_ADDRESS
[▶ 451] | Get | Ausgewählte, einzelne Subscriber Multicast-Adresse des Netzwerkadapters. Wenn gesetzt, empfängt die Netzwerkadapterbausteininstanz nur die Subscriber-Nachrichten, die exakt an diese Adresse gesendet werden. In diesem Fall wird die Adresse als Filter für die Subscriber-Nachrichten verwendet. Alle anderen Nachrichten werden entweder an das Betriebssystem oder (wenn vorhanden) an weitere Instanzen des Netzwerkadapterbausteins weitergeleitet.

Hinweis Ab der TwinCAT TF6510 IEC 61850 Telecontrol v3.1.98.0 und höher: Wenn diese Adresse nicht gesetzt wurde ('00-00-00-00-00-00'), dann werden als Filter für die Subscriber-Nachrichten alle Multicast-Adressen und APPID's verwendet, die an den Subscriber Steuerblockinstanzen konfiguriert wurden. In diesem Fall können z.B. in den GOOSE-Steuerblockinstanzen unterschiedliche Multicast-Adressen konfiguriert werden. Die konfigurierten Subscriber APPID's (wenn verwendet > 0), müssen den APPID's auf der Publisher-Seite entsprechen. Alle empfangenen Nachrichten, die keinem Steuerblock zugeordnet werden können, werden an das Betriebssystem oder an weitere Instanzen des Netzwerkadapterbausteins weitergeleitet. |
| oid | OTCID
[▶ 452] | Get | TwinCAT-Object-ID des Netzwerkadapters. |
| sMacAddr | STRING(17) | Get | Textrepräsentation der MAC-Adresse des Netzwerkadapters. |
| sMulticastAddr | STRING(17) | Get
Set | Textrepräsentation der Multicast-Adresse des Netzwerkadapters. |

6.5.2.2 I_MmsClientSettingsClass

Objekte, die diese Schnittstelle implementieren, dienen der Konfiguration der MMS-Kommunikationsschicht.

Namensraum: Tc3_Mms [▶ 97]

Bibliothek: Tc3_Mms (Tc3_Mms.compiled-library)

Vererbungshierarchie

I_SocketClientSettingsClass [▶ 329] -> I_TpktClientSettingsClass [▶ 334] -> I_Rfc1006ClientSettingsClass [▶ 327] -> I_UlosiClientSettingsClass [▶ 335] -> I_MmsClientSettingsClass

```
INTERFACE I_MmsClientSettingsClass EXTENDS I_UlosiClientSettingsClass
```

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|-------------------------|---|---------|---|
| nLocalDetail | T_MmsInteger32 [▶ 478] | Get | MMS Initiate-RequestPDU-Parameter: localDetailCalling. |
| nMaxServOutCalling | T_MmsInteger16 [▶ 477] | Get | MMS Initiate-RequestPDU-Parameter: proposedMaxServOutstandingCalling |
| nMaxServOutCalled | T_MmsInteger16 [▶ 477] | Get | MMS Initiate-RequestPDU-Parameter: proposedMaxServOutstandingCalled. |
| nNestingLevel | T_MmsInteger8 [▶ 477] | Get | MMS Initiate-RequestPDU-Parameter: proposedDataStructureNestingLevel. |
| nMmsVersionNumber | T_MmsInteger16 [▶ 477] | Get | MMS initRequestDetail-Parameter: proposedVersionNumber. |
| ipParameters | I_MmsParameterSupportOptions [▶ 325] | Get | Schnittstellenzeiger vom Objekt zur Konfiguration der MMS initRequestDetail-Parameter: proposedParameterCBB. |
| ipServices | I_MmsServiceSupportOptions [▶ 324] | Get | Schnittstellenzeiger vom Objekt zur Konfiguration der MMS initRequestDetail-Parameter: servicesSupportedCalling. |
| ipExtendedServices | I_MmsAdditionalSupportOptions [▶ 325] | Get | Schnittstellenzeiger vom Objekt zur Konfiguration der MMS initRequestDetail-Parameter: additionalSupportedCalling. |
| ipExtendedParameters | I_MmsAdditionalCBBOptions [▶ 326] | Get | Schnittstellenzeiger vom Objekt zur Konfiguration der MMS initRequestDetail-Parameter: additionalCbbSupportedCalling. |
| sPrivilegeClassIdentity | T_MmsVisibleString [▶ 478] | Get | MMS-initRequestDetail-Parameter: privilegeClassIdentityCalled. |
| tMmsTimeout | TIME | Get | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceMms | E_TraceLevel [▶ 449] | Get | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |

6.5.2.2.1 I_MmsServiceSupportOptions

Konfiguriert die unterstützten Dienste der MMS-Kommunikationsschicht.

Vererbungshierarchie

I_MmsServiceSupportOptions

```
INTERFACE I_MmsServiceSupportOptions
```


 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|---------------------------------|------|---------|--------------|
| bCancel | BOOL | Get,Set | |
| bConclude | BOOL | Get,Set | |
| bDefineNamedType | BOOL | Get,Set | |
| bDefineNamedVariable | BOOL | Get,Set | |
| bDefineNamedVariableList | BOOL | Get,Set | |
| bDefineScatteredAccess | BOOL | Get,Set | |
| bDeleteNamedType | BOOL | Get,Set | |
| bDeleteNamedVariableList | BOOL | Get,Set | |
| bDeleteVariableAccess | BOOL | Get,Set | |
| bGetDomainAttributes | BOOL | Get,Set | |
| bGetNamedTypeAttributes | BOOL | Get,Set | |
| bGetNamedVariableListAttributes | BOOL | Get,Set | |
| bGetNameList | BOOL | Get,Set | |
| bGetScatteredAccessAttributes | BOOL | Get,Set | |
| bGetVariableAccessAttributes | BOOL | Get,Set | |
| bIdentify | BOOL | Get,Set | |
| bInformationReport | BOOL | Get,Set | |
| bRead | BOOL | Get,Set | |
| bRename | BOOL | Get,Set | |
| bStatus | BOOL | Get,Set | |
| bWrite | BOOL | Get,Set | |

6.5.2.2 I_MmsParameterSupportOptions

Konfiguriert die unterstützten Parameter der MMS-Kommunikationsschicht.

Vererbungshierarchie

I_MmsParameterSupportOptions

INTERFACE I_MmsParameterSupportOptions

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|-------|------|---------|--|
| bCspi | BOOL | Get,Set | |
| bCsr | BOOL | Get,Set | |
| bStr1 | BOOL | Get,Set | Array-Support |
| bStr2 | BOOL | Get,Set | Struktur-Supprt |
| bVadr | BOOL | Get,Set | |
| bValt | BOOL | Get,Set | Support für Index und Index-Range-Werte (alternate access) |
| bVlis | BOOL | Get,Set | Support für namedVariableList |
| bVnam | BOOL | Get,Set | Support für namedVariables |

6.5.2.2.3 I_MmsAdditionalSupportOptions

Konfiguriert die zusätzlich unterstützten MMS-Dienste.

Vererbungshierarchie

I_MmsAdditionalSupportOptions

```
INTERFACE I_MmsAdditionalSupportOptions
```

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|-----------|------|----------|--------------|
| bVmdReset | BOOL | Get,Set | |
| bVmdStop | BOOL | Get,Set | |
| bSelect | BOOL | Get, Set | |

6.5.2.2.4 I_MmsAdditionalCBBOptions

Konfiguriert die zusätzlich unterstützten MMS CBB-Parameter.

Vererbungshierarchie

I_MmsAdditionalCBBOptions

```
INTERFACE I_MmsAdditionalCBBOptions
```

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|------|------|----------|--------------|
| bDei | BOOL | Get,Set | |
| bDes | BOOL | Get,Set | |
| bRed | BOOL | Get, Set | |

6.5.2.3 I_MmsServerSettingsClass

Objekte, die diese Schnittstelle implementieren, dienen der Konfiguration der MMS-Kommunikationsschicht.

Namensraum: [Tc3_Mms](#) [[▶ 97](#)]

Bibliothek: Tc3_Mms (Tc3_Mms.compiled-library)

Vererbungshierarchie

[I_SocketServerSettingsClass](#) [[▶ 330](#)] -> [I_TpktServerSettingsClass](#) [[▶ 335](#)] -> [I_Rfc1006ServerSettingsClass](#) [[▶ 328](#)] -> [I_UlosiServerSettingsClass](#) [[▶ 336](#)] -> [I_MmsServerSettingsClass](#)

```
INTERFACE I_MmsServerSettingsClass EXTENDS I_UlosiServerSettingsClass
```

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|-------------------------|---|---------|---|
| nLocalDetail | T_MmsInteger32 > 478 | Get | MMS Initiate-RequestPDU-Parameter: localDetailCalling. |
| nMaxServOutCalling | T_MmsInteger16 > 477 | Get | MMS Initiate-RequestPDU-Parameter: proposedMaxServOutstandingCalling |
| nMaxServOutCalled | T_MmsInteger16 > 477 | Get | MMS Initiate-RequestPDU-Parameter: proposedMaxServOutstandingCalled. |
| nNestingLevel | T_MmsInteger8 > 477 | Get | MMS Initiate-RequestPDU-Parameter: proposedDataStructureNestingLevel. |
| nMmsVersionNumber | T_MmsInteger16 > 477 | Get | MMS initRequestDetail-Parameter: proposedVersionNumber. |
| ipParameters | I_MmsParameterSupportOptions > 325 | Get | Schnittstellenzeiger vom Objekt zur Konfiguration der MMS initRequestDetail-Parameter: proposedParameterCBB. |
| ipServices | I_MmsServiceSupportOptions > 324 | Get | Schnittstellenzeiger vom Objekt zur Konfiguration der MMS initRequestDetail-Parameter: servicesSupportedCalling. |
| ipExtendedServices | I_MmsAdditionalSupportOptions > 325 | Get | Schnittstellenzeiger vom Objekt zur Konfiguration der MMS initRequestDetail-Parameter: additionalSupportedCalling. |
| ipExtendedParameters | I_MmsAdditionalCBBOptions > 326 | Get | Schnittstellenzeiger vom Objekt zur Konfiguration der MMS initRequestDetail-Parameter: additionalCbbSupportedCalling. |
| sPrivilegeClassIdentity | T_MmsVisibleString > 478 | Get | MMS-initRequestDetail-Parameter: privilegeClassIdentityCalled. |
| tMmsTimeout | TIME | Get | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceMms | E_TraceLevel > 449 | Get | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |

6.5.2.4 I_Rfc1006ClientSettingsClass

Objekte, die diese Schnittstelle implementieren, dienen der Konfiguration der RFC1006-Kommunikationsschicht.

Namensraum: [Tc3_Rfc1006](#) [|> 97](#)

Bibliothek: [Tc3_Rfc1006](#) ([Tc3_Rfc1006.compiled-library](#))

Vererbungshierarchie

[I_SocketClientSettingsClass](#) [|> 329](#) -> [I_TpktClientSettingsClass](#) [|> 334](#) -> [I_Rfc1006ClientSettingsClass](#)

```
INTERFACE I_Rfc1006ClientSettingsClass EXTENDS I_TpktClientSettingsClass
```

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|---------------------|--|---------|--|
| sCalling_T_Selector | STRING | Get | RFC 1006 Parameter: Calling transport address selector (source). |
| sCalled_T_Selector | STRING | Get | RFC 1006 Parameter: Called transport address selector (destination). |
| eTpduSize | E_Rfc1006TpduSize
[▶ 448] | Get | Maximale TPDU-Bytelänge. |
| tRfc1006Timeout | TIME | Get | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceRfc1006 | E_TraceLevel [▶ 449] | Get | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |

6.5.2.5 I_Rfc1006ServerSettingsClass

Objekte, die diese Schnittstelle implementieren, dienen der Konfiguration der RFC1006-Kommunikationsschicht.

Namensraum: [Tc3_Rfc1006](#) [▶ 97]

Bibliothek: Tc3_Rfc1006 (Tc3_Rfc1006.compiled-library)

Vererbungshierarchie

[I_SocketServerSettingsClass](#) [▶ 330] -> [I_TpktServerSettingsClass](#) [▶ 335] -> [I_Rfc1006ServerSettingsClass](#)

```
INTERFACE I_Rfc1006ServerSettingsClass EXTENDS I_TpktServerSettingsClass
```

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|---------------------|--|---------|--|
| sCalling_T_Selector | STRING | Get | RFC 1006 Parameter: Calling transport address selector (source). |
| sCalled_T_Selector | STRING | Get | RFC 1006 Parameter: Called transport address selector (destination). |
| eTpduSize | E_Rfc1006TpduSize
[▶ 448] | Get | Maximale TPDU-Bytelänge. |
| tRfc1006Timeout | TIME | Get | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |

6.5.2.6 I_ScsmClientSettingsClass

Objekte, die diese Schnittstelle implementieren, dienen der Konfiguration der SCSM-Kommunikationsschicht (Specific Communication Service Mapping).

Namensraum: [Tc3_iec61850_8_1](#) [▶ 97]

Bibliothek: Tc3_iec61850_8_1 (Tc3_iec61850_8_1.compiled-library)

Vererbungshierarchie

[I_SocketClientSettingsClass](#) [▶ 329] -> [I_TpktClientSettingsClass](#) [▶ 334] -> [I_Rfc1006ClientSettingsClass](#) [▶ 327] -> [I_UlosiClientSettingsClass](#) [▶ 335] -> [I_MmsClientSettingsClass](#) [▶ 323] -> [I_ScsmClientSettingsClass](#)

```
INTERFACE I_ScsmClientSettingsClass EXTENDS I_MmsClientSettingsClass
```

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|-----------------------------------|---------------------------------------|---------|--|
| bGetDataSetDirectoryMembersReload | BOOL | Get | Aktiviert/deaktiviert die automatische Initialisierung der Datasets/Dataset-Member. |
| eScsmEdition | E_ScsmEdition [▶ 448] | Get | IEC 61850 Datenmodell Edition (1 oder 2) |
| tScsmTimeout | TIME | Get | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceScsm | T_TraceLevel [▶ 449] | Get | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |

6.5.2.7 I_ScsmServerSettingsClass

Objekte, die diese Schnittstelle implementieren, dienen der Konfiguration der SCSM-Kommunikationsschicht (Specific Communication Service Mapping).

Namensraum: [Tc3_iec61850_8_1 \[▶ 97\]](#)

Bibliothek: Tc3_iec61850_8_1 (Tc3_iec61850_8_1.compiled-library)

Vererbungshierarchie

[I_SocketServerSettingsClass \[▶ 330\]](#) -> [I_TpktServerSettingsClass \[▶ 335\]](#) -> [I_Rfc1006ServerSettingsClass \[▶ 328\]](#) -> [I_UlosiServerSettingsClass \[▶ 336\]](#) -> [I_MmsServerSettingsClass \[▶ 326\]](#) -> [I_ScsmServerSettingsClass](#)

```
INTERFACE I_ScsmServerSettingsClass EXTENDS I_MmsServerSettingsClass
```

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|-----------------------------------|---------------------------------------|---------|--|
| bGetDataSetDirectoryMembersReload | BOOL | Get | Aktiviert/deaktiviert die automatische Initialisierung der Datasets/Dataset-Member. |
| eScsmEdition | E_ScsmEdition [▶ 448] | Get | IEC 61850 Datenmodell Edition (1 oder 2) |
| tScsmTimeout | TIME | Get | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceScsm | T_TraceLevel [▶ 449] | Get | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |

6.5.2.8 I_SocketClientSettingsClass

Objekte, die diese Schnittstelle implementieren, dienen der Konfiguration der TCP/IP-Kommunikationsschicht.

Namensraum: [Tc3_Sockets \[▶ 97\]](#)

Bibliothek: Tc3_Sockets (Tc3_Sockets.compiled-library)

Vererbungshierarchie

[I_SocketClientSettingsClass](#) -> [__System.IQueryInterface](#)

```
INTERFACE I_SocketClientSettingsClass EXTENDS __System.IQueryInterface
```

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|----------------|--|---------|--|
| bAutoCleanup | BOOL | Get | Aktiviert/Deaktiviert das automatische Aufräumen/Schließen der unbenutzten Sockets (z.B. nach Programmdownload oder Reset der SPS). |
| sSrvNetID | T_AmsNetID
[▶ 452] | Get | TwinCAT Netzwerkadresse des TwinCAT TCP/IP Servers (TF6310). Beim Leerstring wird die Netzwerkadresse des lokalen TwinCAT Systems verwendet. |
| nRemotePort | UDINT | Get | TCP/IP Portnummer des Servers. |
| sRemoteHost | T_Ipv4Addr
[▶ 452] | Get | IPv4-Netzwerkadresse des Servers. |
| tConnect | TIME | Get | Früheste Zeit, nach deren Ablauf eine neue Verbindung aufgebaut werden kann. |
| tAdsTimeout | TIME | Get | Maximale Zeit, die bei der Ausführung der ADS-Dienste nicht überschritten werden darf. |
| tSocketTimeout | TIME | Get | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| bThrottleMode | BOOL | Get | Aktiviert/deaktiviert den Empfangsdaten-Throttle-Polling-Mode. |
| throttleTimes | T_ThrottleTimes
[▶ 453] | Get | Konfiguration der Zykluszeiten für den Empfangsdaten-Throttle-Polling-Mode. |
| eTraceSocket | E_TraceLevel
[▶ 449] | Get | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| ipTLS | I_SocketTlsSettingsClass [▶ 331] | Get | Konfigurationsparameter der TLS-Protokollschicht (Transport Layer Security). |

6.5.2.9 I_SocketServerSettingsClass

Objekte, die diese Schnittstelle implementieren, dienen der Konfiguration der TCP/IP-Kommunikationsschicht.

Namensraum: [Tc3_Sockets](#) [▶ 97]

Bibliothek: Tc3_Sockets (Tc3_Sockets.compiled-library)

Vererbungshierarchie

I_SocketServerSettingsClass -> __System.IQueryInterface

```
INTERFACE I_SocketServerSettingsClass EXTENDS __System.IQueryInterface
```

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|----------------|--|---------|---|
| bAutoCleanup | BOOL | Get | Aktiviert/Deaktiviert das automatische Aufräumen/Schließen der unbenutzten Sockets (z.B. nach Programmdownload oder Reset der SPS). |
| bEnable | BOOL | Get | Aktiviert/deaktiviert das Herstellen der Serververbindungen. Beim „TRUE“ wird Listen-Befehl aktiviert (horchen und ankommende Verbindungen werden akzeptiert) und beim „FALSE“ abgelehnt/deaktiviert. |
| sSrvNetID | T_AmsNetID [▶ 452] | Get | TwinCAT Netzwerkadresse des TwinCAT TCP/IP Servers (TF6310). Beim Leerstring wird die Netzwerkadresse des lokalen TwinCAT Systems verwendet. |
| nLocalPort | UDINT | Get | TCP/IP Portnummer des Servers. |
| sLocalHost | T_Ipv4Addr [▶ 452] | Get | IPv4-Netzwerkadresse des Servers. |
| tListen | TIME | Get | Wiederholungszykluszeit für den internen TCP/IP-Socket Listen-Befehl. |
| tAccept | TIME | Get | Wiederholungszykluszeit für die internen TCP/IP-Socket Accept-Befehle. |
| tAdsTimeout | TIME | Get | Maximale Zeit, die bei der Ausführung der ADS-Dienste nicht überschritten werden darf. |
| tSocketTimeout | TIME | Get | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| bThrottleMode | BOOL | Get | Aktiviert/deaktiviert den Empfangsdaten-Throttle-Polling-Mode. |
| throttleTimes | T_ThrottleTimes [▶ 453] | Get | Konfiguration der Zykluszeiten für den Empfangsdaten-Throttle-Polling-Mode. |
| eTraceSocket | E_TraceLevel [▶ 449] | Get | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |
| ipTLS | I_SocketTlsSettingsClass [▶ 331] | Get | Konfigurationsparameter der TLS-Protokollschicht (Transport Layer Security). |

6.5.2.10 I_SocketTlsSettingsClass

Objekte, die diese Schnittstelle implementieren, dienen der Konfiguration der TLS (Transport Layer Security) Kommunikationsschicht.

Namensraum: [Tc3_Sockets](#) [[▶ 97](#)]

Bibliothek: Tc3_Sockets (Tc3_Sockets.compiled-library)

Vererbungshierarchie

I_SocketTlsSettingsClass -> [__System.IQueryInterface](#)

```
INTERFACE I_SocketTlsSettingsClass EXTENDS __System.IQueryInterface
```

Methoden

| Name | Beschreibung |
|---|---|
| AddCa [▶ 332] | Speichert den Dateipfad des CA-Zertifikats in den TLS-Konfigurationseinstellungen. |
| AddCert [▶ 333] | Speichert den Dateipfad des Client- oder Server-Zertifikats in den TLS-Konfigurationseinstellungen. |
| AddCrl [▶ 333] | Speichert den Dateipfad der Zertifikatsperrliste (Certificate Revocation List, CRL) in den TLS-Konfigurationseinstellungen. |
| AddPsk [▶ 333] | Speichert den PSK-Key und PSK-Identity-String in den TLS-Konfigurationseinstellungen. |
| Reset [▶ 334] | Setzt alle TLS-Konfigurationseinstellungen zurück. |
| SetConnectFlags [▶ 334] | Speichert zusätzliche, optionale TLS-Konfigurationseinstellungen des Clients. |
| SetListenFlags [▶ 334] | Speichert zusätzliche, optionale TLS-Konfigurationseinstellungen des Servers. |

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------------|--|---------|--|
| nSecurePort | UDINT | Get | Sichere ISO TP0 (IANA) Portnummer für die TLS-Kommunikation. |
| connectFlags | ST_TlsConnectFlags | Get | Zusätzliche TLS-Konfigurationseinstellungen des Clients. |
| listenFlags | ST_TlsListenFlags | Get | Zusätzliche TLS-Konfigurationseinstellungen des Servers. |
| pskKey | PVOID | Get | Zeigt auf das erste Byte des PSK-Schlüssels. |
| pskKeyLen | UDINT(0..TCPADS_TLS_MAX_PSK_KEY_SIZE) | Get | Bytelänge des PSK-Schlüssels. |
| sCaPath | STRING(TCPADS_TLS_CERTIFICATE_PATH_SIZE) | Get | Dateipfad des CA-Zertifikats. |
| sCertPath | STRING(TCPADS_TLS_CERTIFICATE_PATH_SIZE) | Get | Dateipfad des Client- oder Server-Zertifikats. |
| sCrlPath | STRING(TCPADS_TLS_CERTIFICATE_PATH_SIZE) | Get | Dateipfad von der Zertifikatsperrliste (CRL). |
| sIdentity | STRING(TCPADS_TLS_PSK_IDENTITY_SIZE) | Get | Identity-String des PSK-Schlüssels. |
| sKeyPath | STRING(TCPADS_TLS_CERTIFICATE_PATH_SIZE) | Get | Dateipfad des privaten Schlüssels. |
| sKeyPwd | STRING(TCPADS_TLS_KEY_PASSWORD_SIZE) | Get | Passwort-String des Zertifikats. |

6.5.2.10.1 AddCa

| AddCa | |
|---------|--|
| sCaPath | STRING(TCPADS_TLS_CERTIFICATE_PATH_SIZE) |

I_SocketTlsSettingsClass AddCa

Diese Methode speichert den Dateipfad des CA Zertifikats in den TLS-Konfigurationseinstellungen.

```
METHOD FINAL AddCa : I_SocketTlsSettingsClass
VAR_INPUT
    sCaPath : STRING(TCPADS_TLS_CERTIFICATE_PATH_SIZE);
END_VAR
```

sCaPath: Dateipfad des CA Zertifikats.

Rückgabeparameter: Eigener Schnittstellenzeiger vom Typ [I_SocketTlsSettingsClass](#) [► 331].

6.5.2.10.2 AddCert

```

AddCert
sCertPath STRING(TCPADS_TLS_CERTIFICATE_PATH_SIZE) I_SocketTlsSettingsClass AddCert
sKeyPath  STRING(TCPADS_TLS_CERTIFICATE_PATH_SIZE)
sKeyPwd  STRING(TCPADS_TLS_KEY_PASSWORD_SIZE)

```

Diese Methode speichert den Dateipfad des Server- oder Client-Zertifikats in den TLS-Konfigurationseinstellungen.

```

METHOD FINAL AddCert : I_SocketTlsSettingsClass
VAR_INPUT
    sCertPath : STRING(TCPADS_TLS_CERTIFICATE_PATH_SIZE);
    sKeyPath  : STRING(TCPADS_TLS_CERTIFICATE_PATH_SIZE);
    sKeyPwd   : STRING(TCPADS_TLS_KEY_PASSWORD_SIZE);
END_VAR

```

sCertPath: Dateipfad des Server- oder Client-Zertifikats.

sKeyPath: Dateipfad des privaten Schlüssels.

sKeyPwd: Passwort-String des Zertifikats.

Rückgabeparameter: Eigener Schnittstellenzeiger vom Typ [I_SocketTlsSettingsClass](#) [► 331].

6.5.2.10.3 AddCrl

```

AddCrl
sCrlPath STRING(TCPADS_TLS_CERTIFICATE_PATH_SIZE) I_SocketTlsSettingsClass AddCrl

```

Diese Methode speichert den Dateipfad der Zertifikatsperrliste (englisch certificate revocation list, CRL) in den TLS-Konfigurationseinstellungen.

```

METHOD FINAL AddCrl : I_SocketTlsSettingsClass
VAR_INPUT
    sCrlPath : STRING(TCPADS_TLS_CERTIFICATE_PATH_SIZE);
END_VAR

```

sCrlPath: Dateipfad der Zertifikatsperrliste (englisch certificate revocation list, CRL).

Rückgabeparameter: Eigener Schnittstellenzeiger vom Typ [I_SocketTlsSettingsClass](#) [► 331].

6.5.2.10.4 AddPsk

```

AddPsk
key POINTER TO BYTE I_SocketTlsSettingsClass AddPsk
sIdentity STRING(TCPADS_TLS_PSK_IDENTITY_SIZE)

```

Diese Methode speichert den PSK (pre-shared key) und den PSK Identity-String in den TLS-Konfigurationseinstellungen.

```

METHOD FINAL AddPsk : I_SocketTlsSettingsClass
VAR_IN_OUT
    key : ARRAY[*] OF BYTE;
END_VAR
VAR_INPUT
    sIdentity : STRING(TCPADS_TLS_PSK_IDENTITY_SIZE);
END_VAR

```

key: Datenpuffer mit dem PSK-Schlüssel (Array mit variabler Länge).

sIdentity: PSK Identity-String.

Rückgabeparameter: Eigener Schnittstellenzeiger vom Typ [I_SocketTlsSettingsClass](#) [► 331].

6.5.2.10.5 Reset

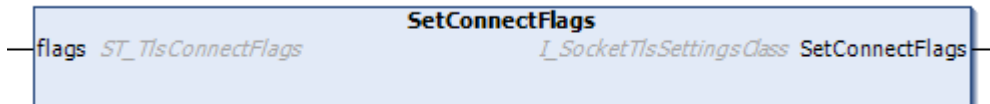


Diese Methode setzt alle TLS-Konfigurationseinstellungen zurück.

```
METHOD FINAL Reset : I_SocketTlsSettingsClass
```

Rückgabeparameter: Eigener Schnittstellenzeiger vom Typ [I_SocketTlsSettingsClass](#) [► 331].

6.5.2.10.6 SetConnectFlags



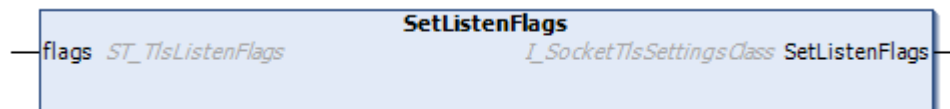
Diese Methode speichert zusätzliche, optionale TLS-Konfigurationseinstellungen des Clients.

```
METHOD FINAL SetConnectFlags : I_SocketTlsSettingsClass
VAR_INPUT
    flags : ST_TlsConnectFlags;
END_VAR
```

flags: Parameter vom Typ `ST_TlsConnectFlags` mit zusätzlichen, optionalen TLS-Konfigurationseinstellungen des Clients. Der Wert der „flags“ Membervariablen „bNoseverCertCheck“ legt fest ob das Server-Zertifikat geprüft wird (FALSE, default) oder nicht geprüft wird (TRUE). Der Wert der „flags“ Membervariablen „blgnoreCnMismatch“ legt fest ob eine Ungleichheit im „CommonName“ des Zertifikats ignoriert wird (TRUE) oder als Fehler gemeldet wird (FALSE, default).

Rückgabeparameter: Eigener Schnittstellenzeiger vom Typ [I_SocketTlsSettingsClass](#) [► 331].

6.5.2.10.7 SetListenFlags



Diese Methode speichert zusätzliche, optionale TLS-Konfigurationseinstellungen des Servers.

```
METHOD FINAL SetListenFlags : I_SocketTlsSettingsClass
VAR_INPUT
    flags : ST_TlsListenFlags;
END_VAR
```

flags: Parameter vom Typ `ST_TlsListenFlags` mit zusätzlichen, optionalen TLS-Konfigurationseinstellungen des Servers. Der Wert der „flags“ Membervariablen „bNoClientCert“ legt fest ob die Konfiguration des Client-Zertifikats erforderlich ist (FALSE, default) oder nicht erforderlich ist (TRUE).

Rückgabeparameter: Eigener Schnittstellenzeiger vom Typ [I_SocketTlsSettingsClass](#) [► 331].

6.5.2.11 I_TpktClientSettingsClass

Objekte, die diese Schnittstelle implementieren, dienen der Konfiguration der TPKT-Kommunikationsschicht (ISO transport services on top of the TCP).

Namensraum: `Tc3_Tpkt` [► 97]

Bibliothek: `Tc3_Tpkt` (`Tc3_Tpkt.compiled-library`)

Vererbungshierarchie

[I_SocketClientSettingsClass](#) [► 329] -> `I_TpktClientSettingsClass`

INTERFACE I_TpktClientSettingsClass EXTENDS I_SocketClientSettingsClass

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------------|--------------------------------------|---------|--|
| tTpktTimeout | TIME | Get | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceTpkt | E_TraceLevel [▶ 449] | Get | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |

6.5.2.12 I_TpktServerSettingsClass

Objekte, die diese Schnittstelle implementieren, dienen der Konfiguration der TPKT-Kommunikationsschicht (ISO transport services on top of the TCP).

Namensraum: [Tc3_Tpkt \[▶ 97\]](#)

Bibliothek: Tc3_Tpkt (Tc3_Tpkt.compiled-library)

Vererbungshierarchie

[I_SocketServerSettingsClass \[▶ 330\]](#) -> [I_TpktServerSettingsClass](#)

INTERFACE I_TpktServerSettingsClass EXTENDS I_SocketServerSettingsClass

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|--------------|--------------------------------------|---------|--|
| tTpktTimeout | TIME | Get | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceTpkt | E_TraceLevel [▶ 449] | Get | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |

6.5.2.13 I_UlosiClientSettingsClass

Objekte, die diese Schnittstelle implementieren, dienen der Konfiguration der ULOSI-Kommunikationsschicht (Upper Layer OSI).

Namensraum: [Tc3_Ulosi \[▶ 97\]](#)

Bibliothek: Tc3_Ulosi (Tc3_Ulosi.compiled-library)

Vererbungshierarchie

[I_SocketClientSettingsClass \[▶ 329\]](#) -> [I_TpktClientSettingsClass \[▶ 334\]](#) -> [I_Rfc1006ClientSettingsClass \[▶ 327\]](#) -> [I_UlosiClientSettingsClass](#)

INTERFACE I_UlosiClientSettingsClass EXTENDS I_Rfc1006ClientSettingsClass

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|-----------------------|--|---------|--|
| sCalling_S_Selector | STRING | Get | Calling-S-selector. |
| sCalled_S_Selector | STRING | Get | Called-S-selector. |
| nSessionRequirement | WORD | Get | Session requirements. |
| sCalling_P_Selector | STRING | Get | Calling-P-selector. |
| sCalled_P_Selector | STRING | Get | Called-P-selector. |
| nProtocol_Version | BYTE | Get | Protocol version. |
| sContext_Name | STRING | Get | Application context name (object identifier, dotted presentation), MMS. |
| sCalled_AP_Title | STRING | Get | Called-AP-title. |
| nCalled_AE_Qualifier | UDINT | Get | Called-AE-qualifier. |
| nCalled_AP_InvID | UDINT | Get | Called-AP-invocation-identifier. |
| nCalled_AE_InvID | UDINT | Get | Called-AE-invocation-identifier. |
| sCalling_AP_Title | STRING | Get | Calling-AP-title. |
| nCalling_AE_Qualifier | UDINT | Get | Calling-AE-qualifier. |
| nCalling_AP_InvID | UDINT | Get | Calling-AP-invocation-identifier. |
| nCalling_AE_InvID | UDINT | Get | Calling-AE-invocation-identifier. |
| nRequirements | BYTE | Get | Requirements (1 == authentication). |
| sMechanism_Name | STRING | Get | Authentication mechanism name. |
| sAuthent_Value | STRING | Get | Authentication value (password). |
| tUlosiTimeout | TIME | Get | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceUlosi | E_TraceLevel [▶ 449] | Get | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |

6.5.2.14 I_UlosiServerSettingsClass

Objekte, die diese Schnittstelle implementieren, dienen der Konfiguration der ULOSI-Kommunikationsschicht (Upper Layer OSI).

Namensraum: [Tc3_Ulosi](#) [[▶ 97](#)]

Bibliothek: [Tc3_Ulosi](#) ([Tc3_Ulosi.compiled-library](#))

Vererbungshierarchie

[I_SocketServerSettingsClass](#) [[▶ 330](#)] -> [I_TpktServerSettingsClass](#) [[▶ 335](#)] -> [I_Rfc1006ServerSettingsClass](#) [[▶ 328](#)] -> [I_UlosiServerSettingsClass](#)

```
INTERFACE I_UlosiServerSettingsClass EXTENDS I_Rfc1006ServerSettingsClass
```

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|-----------------------|--|---------|--|
| sCalling_S_Selector | STRING | Get | Calling-S-selector. |
| sCalled_S_Selector | STRING | Get | Called-S-selector. |
| nSessionRequirement | WORD | Get | Session requirements. |
| sCalling_P_Selector | STRING | Get | Calling-P-selector. |
| sCalled_P_Selector | STRING | Get | Called-P-selector. |
| nProtocol_Version | BYTE | Get | Protocol version. |
| sContext_Name | STRING | Get | Application context name (object identifier, dotted presentation), MMS. |
| sCalled_AP_Title | STRING | Get | Called-AP-title. |
| nCalled_AE_Qualifier | UDINT | Get | Called-AE-qualifier. |
| nCalled_AP_InvID | UDINT | Get | Called-AP-invocation-identifier. |
| nCalled_AE_InvID | UDINT | Get | Called-AE-invocation-identifier. |
| sCalling_AP_Title | STRING | Get | Calling-AP-title. |
| nCalling_AE_Qualifier | UDINT | Get | Calling-AE-qualifier. |
| nCalling_AP_InvID | UDINT | Get | Calling-AP-invocation-identifier. |
| nCalling_AE_InvID | UDINT | Get | Calling-AE-invocation-identifier. |
| nRequirements | BYTE | Get | Requirements (1 == authentication). |
| sMechanism_Name | STRING | Get | Authentication mechanism name. |
| sAuthent_Value | STRING | Get | Authentication value (password). |
| tUlosiTimeout | TIME | Get | Maximale Zeit, die bei der Ausführung der bestätigten Dienste nicht überschritten werden darf. |
| eTraceUlosi | E_TraceLevel [▶ 449] | Get | Konfiguriert die Prioritätsstufe der Protokollierung der Fehlermeldungen und Logmeldungen. |

6.5.3 Kommunikation

6.5.3.1 GOOSE

6.5.3.1.1 I_GseAdapterClass

Schnittstellenzeiger von diesem Typ identifizieren für die GSE-Kommunikation nutzbare Netzwerkadapter.

Namensraum: [Tc3_Gse](#) [[▶ 97](#)]

Bibliothek: Tc3_Gse (Tc3_Gse.compiled-library)

Vererbungshierarchie

I_GseAdapterClass

INTERFACE I_GseAdapterClass

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|------------|---|---------|--|
| ipSettings | I_GseAdapterSettingsClasses [▶ 322] | Get | Protokolleinstellungen des GSE-Adapters. |

6.5.3.1.2 I_GseGoCBImplClass

Objekte, die diese Schnittstelle implementieren, sind Steuerblockinstanzen für Generic Object Oriented Substation Events (GOOSE). Der TwinCAT Telecontrol Configurator generiert für jeden konfigurierten GOOSE-Steuerblock, eine Instanz des Funktionsbausteins [FB_GseGoCBImplClass \[▶ 241\]](#), der diese Schnittstelle implementiert.

Namensraum: [Tc3_Gse \[▶ 97\]](#)

Bibliothek: Tc3_Gse (Tc3_Gse.compiled-library)

Vererbungshierarchie

[I_AcsiCommonNodeClass \[▶ 320\]](#) -> [I_AcsiCommonDataClass \[▶ 311\]](#) -> [I_AcsiCommonControlBlockClass \[▶ 311\]](#) -> [I_AcsiCommonGooseControlBlockClass \[▶ 315\]](#) -> [I_ScsmGoCBImplClass \[▶ 346\]](#) -> [I_GseGoCBImplClass](#)

```
INTERFACE I_GseGoCBImplClass EXTENDS I_ScsmGoCBImplClass
```



Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------------|--|---------|----------------------------------|
| ipPublisher | I_GseGoCBPublisherClass [▶ 338] | Get | Implementierung des Publishers. |
| ipSubscriber | I_GseGoCBSubscriberClass [▶ 339] | Get | Implementierung des Subscribers. |

6.5.3.1.3 I_GseGoCBPublisherClass

Objekte, die diese Schnittstelle implementieren, steuern das Verhalten eines GOOSE-Publishers.

Namensraum: [Tc3_Gse \[▶ 97\]](#)

Bibliothek: Tc3_Gse (Tc3_Gse.compiled-library)

Vererbungshierarchie

[I_ServiceErrorClass \[▶ 392\]](#) -> [I_GseGoCBPublisherClass](#)

```
INTERFACE I_GseGoCBPublisherClass EXTENDS I_ServiceErrorClass
```



Methoden

| Name | Beschreibung |
|---------------------------------|--|
| Execute [▶ 338] | Führt die Publisher-Zustandsmaschine aus. |
| Start [▶ 339] | Startet das Versenden von GOOSE-Nachrichten. |
| Stop [▶ 339] | Stoppt das Versenden von GOOSE-Nachrichten. |
| Update [▶ 339] | Löst ein sofortiges Versenden einer GOOSE-Nachricht aus. |

6.5.3.1.3.1 Execute

Führt die Publisher-Zustandsmaschine aus.

```
METHOD Execute : BOOL
VAR_OUTPUT
    ipError : I_ServiceErrorClass;
END_VAR
```

ipError: Schnittstellenzeiger vom Typ [I_ServiceErrorClass \[▶ 392\]](#). Über diesen Zeiger kann bei einer negativen Rückmeldung der Fehlercode, Fehlerquelle oder Fehlertext abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) beim Erfolg oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.1.3.2 Start

Startet das Versenden von GOOSE-Nachrichten.

```
METHOD Start : BOOL
VAR_INPUT
    ipAdapter : I_GseAdapterClass;
END_VAR
VAR_OUTPUT
    ipError : I_ServiceErrorClass;
END_VAR
```

ipAdapter: Schnittstellenzeiger vom Typ [I_GseAdapterClass](#) [► 337]. Dieser Zeiger identifiziert den Netzwerkadapter, der fürs Versenden der GOOSE-Nachrichten verwendet werden soll.

ipError: Schnittstellenzeiger vom Typ [I_ServiceErrorClass](#) [► 392]. Über diesen Zeiger kann bei einer negativen Rückmeldung der Fehlercode, Fehlerquelle oder Fehlertext abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) beim Erfolg oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.1.3.3 Stop

Stoppt das Versenden von GOOSE-Nachrichten.

```
METHOD Stop : BOOL
VAR_OUTPUT
    ipError : I_ServiceErrorClass;
END_VAR
```

ipError: Schnittstellenzeiger vom Typ [I_ServiceErrorClass](#) [► 392]. Über diesen Zeiger kann bei einer negativen Rückmeldung der Fehlercode, Fehlerquelle oder Fehlertext abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) beim Erfolg oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.1.3.4 Update

Löst ein sofortiges Versenden einer neuen GOOSE-Nachricht aus. Neue Nachrichten werden bei Wertänderungen der GOOSE-Dataset Member versendet. Nachrichtenwiederholungen werden automatisch, im Hintergrund, getriggert durch die Execute-Methode der Publisher-Zustandsmaschine versendet.

```
METHOD Update : BOOL
VAR_OUTPUT
    ipError : I_ServiceErrorClass;
END_VAR
```

ipError: Schnittstellenzeiger vom Typ [I_ServiceErrorClass](#) [► 392]. Über diesen Zeiger kann bei einer negativen Rückmeldung der Fehlercode, Fehlerquelle oder Fehlertext abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) beim Erfolg oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.1.4 I_GseGoCBSubscriberClass

Objekte, die diese Schnittstelle implementieren, steuern das Verhalten eines GOOSE-Subscribers.

Namensraum: [Tc3_Gse](#) [► 97]

Bibliothek: [Tc3_Gse](#) ([Tc3_Gse.compiled-library](#))

Vererbungshierarchie

[I_ServiceErrorClass](#) [► 392] -> [I_GseGoCBSubscriberClass](#)

```
INTERFACE I_GseGoCBSubscriberClass EXTENDS I_ServiceErrorClass
```

Methoden

| Name | Beschreibung |
|---|--|
| Execute [▶ 340] | Führt die Subscriber-Zustandsmaschine aus. |
| Enable [▶ 340] | Startet das Empfangen von GOOSE-Nachrichten. |
| Disable [▶ 340] | Stoppt das Empfangen von GOOSE-Nachrichten. |

6.5.3.1.4.1 Execute

Führt die Subscriber-Zustandsmaschine aus.

```
METHOD Execute : BOOL
VAR_OUTPUT
    ipError : I_ServiceErrorClass;
END_VAR
```

ipError: Schnittstellenzeiger vom Typ [I_ServiceErrorClass](#) [[▶ 392](#)]. Über diesen Zeiger kann bei einer negativen Rückmeldung der Fehlercode, Fehlerquelle oder Fehlertext abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) beim Erfolg oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.1.4.2 Enable

Startet das Empfangen von GOOSE-Nachrichten.

```
METHOD Enable : BOOL
VAR_INPUT
    ipAdapter : I_GseAdapterClass;
END_VAR
VAR_OUTPUT
    ipError : I_ServiceErrorClass;
END_VAR
```

ipAdapter: Schnittstellenzeiger vom Typ [I_GseAdapterClass](#) [[▶ 337](#)]. Dieser Zeiger identifiziert den Netzwerkadapter, der fürs Versenden der GOOSE-Nachrichten verwendet werden soll.

ipError: Schnittstellenzeiger vom Typ [I_ServiceErrorClass](#) [[▶ 392](#)]. Über diesen Zeiger kann bei einer negativen Rückmeldung der Fehlercode, Fehlerquelle oder Fehlertext abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) beim Erfolg oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.1.4.3 Disable

Stoppt das Empfangen von GOOSE-Nachrichten.

```
METHOD Disable : BOOL
VAR_OUTPUT
    ipError : I_ServiceErrorClass;
END_VAR
```

ipError: Schnittstellenzeiger vom Typ [I_ServiceErrorClass](#) [[▶ 392](#)]. Über diesen Zeiger kann bei einer negativen Rückmeldung der Fehlercode, Fehlerquelle oder Fehlertext abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) beim Erfolg oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.1.5 I_GseLinkStatusEventSink

Objekte, die diese Schnittstelle implementieren, besitzen eine benutzerdefinierte „LinkStatusChange“-Dienst Ereignisbehandlungsroutine.

Namensraum: [Tc3_Gse](#) [[▶ 97](#)]


Bibliothek: Tc3_Gse (Tc3_Gse.compiled-library)

Vererbungshierarchie

I_GseLinkStatusEventSink


INTERFACE I_GseLinkStatusEventSink EXTENDS __System.IQueryInterface

 **Methoden zur Ereignisbehandlung**

| Name | Beschreibung |
|--|---|
| OnLinkStatusChange  341 | Benutzerdefinierte „LinkStatusChange“-Dienst Ereignisbehandlungsroutine |

6.5.3.1.5.1 OnLinkStatusChange

```
METHOD OnLinkStatusChange
VAR_INPUT
    ipAdapter : I_GseAdapterClass;
    eStatus   : E_GseLinkStatus;
END_VAR
```

ipAdapter: Schnittstellenzeiger vom Typ: [I_GseAdapterClass](#)  [337](#). Dieser Zeiger gibt die Instanz an, dessen Status sich geändert hat.

eStatus: Der aktuelle Status vom Typ [E_GseLinkStatus](#)  [446](#).

6.5.3.1.6 I_GseSystemClockEventSink

Objekte, die diese Schnittstelle implementieren, besitzen eine benutzerdefinierte SystemClock-Dienst Ereignisbehandlungsroutine für den Zugriff auf eine externe Zeitquelle.

Namensraum: [Tc3_Gse](#)  [97](#)


Bibliothek: Tc3_Gse (Tc3_Gse.compiled-library)

Vererbungshierarchie

I_GseSystemClockEventSink

INTERFACE I_GseSystemClockEventSink EXTENDS __System.IQueryInterface

 **Methoden zur Ereignisbehandlung**

| Name | Beschreibung |
|---|--|
| OnGetSystemTime  341 | Benutzerdefinierte SystemClock-Dienst Ereignisbehandlungsroutine |

6.5.3.1.6.1 OnGetSystemTime

```
METHOD OnGetSystemTime : BOOL
VAR_INPUT
    ipAdapter : I_GseAdapterClass;
END_VAR
VAR_OUTPUT
    tT      : T_UtcTime;
END_VAR
```

ipAdapter: Schnittstellenzeiger vom Typ: [I_GseAdapterClass](#)  [337](#). Dieser Zeiger gibt die Instanz an, die die aktuelle Systemzeit abfragt.

tT: Aktuelle Utc-Time-Zeitinformation vom Typ [T_UtcTime](#)  [479](#).

6.5.3.2 Steuerblock-Zugriff

6.5.3.2.1 I_ScsmBrCBImplClass

Objekte, die diese Schnittstelle implementieren, sind Berichtssteuerblockinstanzen für gepufferte Reports (Buffered Report Control Blocks, BRCBs). Der TwinCAT Telecontrol Configurator generiert Berichtssteuerblockinstanzen, die von [FB_AcsiBufferedReportControlBlock \[▶ 164\]](#) abgeleitet/erweitert werden. Alle Funktionsbausteine, die von [FB_AcsiBufferedReportControlBlock \[▶ 164\]](#) abgeleitet werden, besitzen automatisch eine Default-Implementierung der oben genannten Schnittstelle.

Namensraum: [Tc3_iec61850_8_1 \[▶ 97\]](#)

Bibliothek: [Tc3_iec61850_8_1 \(Tc3_iec61850_8_1.compiled-library\)](#)

Vererbungshierarchie

[I_AcsiCommonNodeClass \[▶ 320\]](#) -> [I_AcsiCommonDataClass \[▶ 311\]](#) -> [I_AcsiCommonControlBlockClass \[▶ 311\]](#) -> [I_AcsiCommonBufferedReportControlBlockClass \[▶ 310\]](#) -> [I_ScsmBrCBImplClass](#)

```
INTERFACE I_ScsmBrCBImplClass EXTENDS I_AcsiCommonBufferedReportControlBlockClass
```

 **Eigenschaften**

| Name | Type | Zugriff | Beschreibung |
|------------------|--|-------------|--|
| bGI | BOOL | Get,
Set | Attributwert: Aktivierung des Generalabfrage-Befehls. |
| bPurgeB
uf | BOOL | Get,
Set | Attributwert: Aktivierung des Purge-Befehls. |
| bRptEna | BOOL | Get,
Set | Attributwert: Aktivierung/Deaktivierung des RCB-Steuerblocks. Weitere Konfiguration benötigt. |
| cOptFlds | ST_AcsiOptionalFields [▶ 459] | Get,
Set | Attributwert: Konfiguration der optionalen Felder, die mit einem Report übertragen werden. |
| cTrgOps | ST_AcsiTriggerConditions [▶ 466] | Get,
Set | Attributwert: Konfiguration der Auslösoptionen für das Senden eines Reports. |
| iResvTm
s | INT | Get,
Set | Attributwert: Konfiguration der Zeit für die Reservierung des RCBs für einen bestimmten Client. |
| nBufTm | DWORD | Get,
Set | Attributwert: Konfiguration der max. Report-Pufferzeit in Millisekunden. |
| nConfRev | DWORD | Get,
Set | Attributwert: Revisionsnummer der Konfiguration des RCB-Steuerblocks. |
| nEntryID | LWORD | Get,
Set | Attributwert: ID des Report-Eintrags als 64-Bit Zahlenwert. |
| nIntgPd | DWORD | Get,
Set | Attributwert: Konfiguration der max. Zeit zwischen den Integrity-Reports in Millisekunden. |
| nSqNum | WORD | Get,
Set | Attributwert: Report-Sequenznummer. |
| oEntryID | T_OCTET8 [▶ 478] | Get,
Set | Attributwert: ID des Report-Eintrags als Oktett-String. |
| oOwner | T_OCTET64 [▶ 479] | Get,
Set | Attributwert: Eigentümer des RCB's. IP-Adresse des Clients welcher den RCB für sich reserviert hat (Offline-/Online-Reservierung). |
| sDatSet | T_AcsiObjectReference [▶ 476] | Get,
Set | Attributwert: Name des verknüpften Data-Sets. |
| sRptID | T_AcsiVisString129 [▶ 476] | Get,
Set | Attributwert: Report-Identifikationsstring. |
| tTimeOfE
ntry | T_BinaryTime [▶ 476] | Get,
Set | Attributwert: Report-Zeitstempel. |
| ipDatSet | I_AcsiCommonDataSetClass [▶ 313] | Get | Schnittstellenzeiger des verknüpften Data-Sets. |
| ipClient | I_ScsmBrCBClientJobClass [▶ 343] | Get | Schnittstellenzeiger der Client-Implementierung des BRCB-Steuerblocks. |
| ipServer | I_ScsmBrCBServerJobClass [▶ 345] | Get | Schnittstellenzeiger der Server-Implementierung des BRCB-Steuerblocks. |

6.5.3.2.2 I_ScsmBrCBClientJobClass

Objekte, die diese Schnittstelle implementieren, steuern das Verhalten des Client-BRCB Steuerblocks.

Namensraum: [Tc3_iec61850_8_1](#) [▶ 97]

Bibliothek: [Tc3_iec61850_8_1](#) (Tc3_iec61850_8_1.compiled-library)

Methoden

| Name | Beschreibung |
|-------------------------------------|--------------------------------------|
| EnableReq [▶ 344] | Aktiviert den Steuerblock. |
| DisableReq [▶ 344] | Deaktiviert den Steuerblock. |
| GReq [▶ 344] | Aktiviert den Generalabfrage-Befehl. |
| PurgeBufReq [▶ 345] | Aktiviert den PurgeBuf-Befehl. |
| ResyncReq [▶ 345] | Aktiviert den Resync-Befehl. |

6.5.3.2.2.1 EnableReq

Aktiviert den BRCB Steuerblock.

```
METHOD FINAL EnableReq : BOOL
VAR_INPUT
    ipClient : I_ScsmClientClass;
    ipDataSet : I_AcsiCommonDataSetClass;
END_VAR
VAR_OUTPUT
    ipResult : I_AsyncServiceResultClass;
END_VAR
```

ipClient: Schnittstellenzeiger vom Typ: [I_ScsmClientClass \[▶ 351\]](#). Dieser Zeiger bestimmt die Dienstausführende Objektinstanz des IEC 61850 Clients.

ipDataSet: Schnittstellenzeiger vom Typ [I_AcsiCommonDataSetClass \[▶ 313\]](#).

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass \[▶ 387\]](#). Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstausführung überwacht und abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.2.2.2 DisableReq

Deaktiviert den BRCB Steuerblock.

```
METHOD FINAL DisableReq : BOOL
VAR_OUTPUT
    ipResult : I_AsyncServiceResultClass;
END_VAR
```

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass \[▶ 387\]](#). Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstausführung überwacht und abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.2.2.3 GReq

Aktiviert den Generalabfrage-Befehl.

```
METHOD FINAL GReq : BOOL
VAR_OUTPUT
    ipResult : I_AsyncServiceResultClass;
END_VAR
```

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass \[▶ 387\]](#). Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstausführung überwacht und abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.2.2.4 PurgeBufReq

Aktiviert den Purge-Befehl.

```
METHOD FINAL PurgeBufReq : BOOL
VAR_INPUT
    ipClient : I_ScsmClientClass;
END_VAR
VAR_OUTPUT
    ipResult : I_AsyncServiceResultClass;
END_VAR
```

ipClient: Schnittstellenzeiger vom Typ: [I_ScsmClientClass](#) [▶ 351]. Dieser Zeiger bestimmt die Dienstausführende Objektinstanz des IEC 61850 Clients.

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass](#) [▶ 387]. Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstausführung überwacht und abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.2.2.5 ResyncReq

Aktiviert den Resync-Befehl.

```
METHOD FINAL ResyncReq : BOOL
VAR_INPUT
    ipClient : I_ScsmClientClass;
    nEntryID : LWORD;
END_VAR
VAR_OUTPUT
    ipResult : I_AsyncServiceResultClass;
END_VAR
```

ipClient: Schnittstellenzeiger vom Typ: [I_ScsmClientClass](#) [▶ 351]. Dieser Zeiger bestimmt die Dienstausführende Objektinstanz des IEC 61850 Clients.

nEntryID: EntryID des Reports für den Puffer-Resynchronisierungsbefehl.

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass](#) [▶ 387]. Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstausführung überwacht und abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.2.3 I_ScsmBrCBServerJobClass

Objekte, die diese Schnittstelle implementieren, steuern das Verhalten des Server-BRCB Steuerblocks.

Namensraum: [Tc3_iec61850_8_1](#) [▶ 97]

Bibliothek: [Tc3_iec61850_8_1](#) (Tc3_iec61850_8_1.compiled-library)

Methoden

| Name | Beschreibung |
|---------------------------------|---|
| Execute [▶ 345] | Führt die RCB-Zustandsmaschine des Servers aus. |

6.5.3.2.3.1 Execute

Führt die RCB-Zustandsmaschine des Servers aus.

```
METHOD Execute : BOOL
```

6.5.3.2.4 I_ScsmGoCBImplClass

Objekte, die diese Schnittstelle implementieren, sind Steuerblockinstanzen für Generic Object Oriented Substation Events (GOOSE). Der TwinCAT Telecontrol Configurator generiert GOOSE-Steuerblockinstanzen, die von [FB_ScsmGoCBImplClass \[▶ 232\]](#) abgeleitet/erweitert werden. Alle Funktionsbausteine, die von [FB_ScsmGoCBImplClass \[▶ 232\]](#) abgeleitet werden, besitzen automatisch eine Default-Implementierung der oben genannten Schnittstelle.

Namensraum: [Tc3_iec61850_8_1 \[▶ 97\]](#)

Bibliothek: [Tc3_iec61850_8_1 \(Tc3_iec61850_8_1.compiled-library\)](#)

Vererbungshierarchie

[I_AcsiCommonNodeClass \[▶ 320\]](#) -> [I_AcsiCommonDataClass \[▶ 311\]](#) -> [I_AcsiCommonControlBlockClass \[▶ 311\]](#) -> [I_AcsiCommonGooseControlBlockClass \[▶ 315\]](#) -> [I_ScsmGoCBImplClass](#)

```
INTERFACE I_ScsmGoCBImplClass EXTENDS I_AcsiCommonGooseControlBlockClass
```



Eigenschaften

| Name | Type | Zugriff | Beschreibung |
|------------|---|----------|--|
| bFixedOffs | BOOL | Get | Attributwert: Aktiviert/Deaktiviert Kodierung mit festen Längen. |
| bGoEna | BOOL | Get, Set | Attributwert: Aktiviert/Deaktiviert den GOOSE-Steuerblock. |
| bNdsCom | BOOL | Get | Attributwert: Gibt an, ob der GOOSE-Steuerblock weitere Konfiguration benötigt. |
| ePRIORITY | E_AcsiVlanPriority [▶ 442] | Get | Prioritätsstufe im VLAN-Netz. |
| ipDatSet | I_AcsiCommonDataSet Class [▶ 313] | Get | Schnittstellenzeiger auf das verknüpfte Data-Set. |
| macAddr | ETHERNET ADDRESS [▶ 451] | Get | Ziel-MAC-Adresse. |
| nAPPID | WORD | Get | Application Identifier. |
| nConfRev | UDINT | Get | Revisionsnummer der Konfiguration des GOOSE-Steuerblocks. |
| nMaxTime | DWORD | Get | Maximale Zeit zwischen den wiederholten Übertragungen einer GOOSE-Nachricht in Millisekunden. |
| nMinTime | DWORD | Get | Zeit zwischen dem direkten Senden nach Datenänderung und der ersten Wiederholung in Millisekunden. |
| nVID | WORD(0..4095) | Get | ID des VLAN-Netzes. |
| oAddr | T_OCTET6 [▶ 478] | Get | Ziel-MAC-Adresse. |
| sAddr | STRING(17) | Get | Ziel-MAC-Adresse. |
| sDatSet | T_AcsiObjectReference [▶ 476] | Get, Set | Name des verknüpften Data-Sets. |
| sGoID | T_AcsiVisString129 [▶ 476] | Get, Set | Die ID der GOOSE-Nachrichten. |

6.5.3.2.5 I_ScsmGoCBClientJobClass

Objekte, die diese Schnittstelle implementieren, steuern das Verhalten des Client-GOCB Steuerblocks.

Namensraum: [Tc3_iec61850_8_1 \[▶ 97\]](#)

Bibliothek: [Tc3_iec61850_8_1 \(Tc3_iec61850_8_1.compiled-library\)](#)

Methoden

| Name | Beschreibung |
|--------------------|------------------------------|
| EnableReq [▶ 347] | Aktiviert den Steuerblock. |
| DisableReq [▶ 347] | Deaktiviert den Steuerblock. |

6.5.3.2.5.1 EnableReq

Aktiviert den GOCB Steuerblock.

```
METHOD FINAL EnableReq : BOOL
VAR_INPUT
    ipClient : I_ScsmClientClass;
    ipDataSet : I_AcsiCommonDataSetClass;
END_VAR
VAR_OUTPUT
    ipResult : I_AsyncServiceResultClass;
END_VAR
```

ipClient: Schnittstellenzeiger vom Typ: [I_ScsmClientClass \[▶ 351\]](#). Dieser Zeiger bestimmt die Dienstausführende Objektinstanz des IEC 61850 Clients.

ipDataSet: Schnittstellenzeiger vom Typ [I_AcsiCommonDataSetClass \[▶ 313\]](#).

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass \[▶ 387\]](#). Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstausführung überwacht und abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.2.5.2 DisableReq

Deaktiviert den GOCB Steuerblock.

```
METHOD FINAL DisableReq : BOOL
VAR_OUTPUT
    ipResult : I_AsyncServiceResultClass;
END_VAR
```

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass \[▶ 387\]](#). Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstausführung überwacht und abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.2.6 I_ScsmGoCBServerJobClass

Objekte, die diese Schnittstelle implementieren, steuern das Verhalten des Server-GOCB Steuerblocks.

Namensraum: [Tc3_iec61850_8_1 \[▶ 97\]](#)

Bibliothek: [Tc3_iec61850_8_1 \(Tc3_iec61850_8_1.compiled-library\)](#)

Methoden

| Name | Beschreibung |
|-----------------|--|
| Execute [▶ 347] | Führt die GOCB-Zustandsmaschine des Servers aus. |

6.5.3.2.6.1 Execute

Führt die GOCB-Zustandsmaschine des Servers aus.

```
METHOD Execute : BOOL
```

6.5.3.2.7 I_ScsmUrCBImplClass

Objekte, die diese Schnittstelle implementieren, sind Berichtssteuerblockinstanzen für nicht gepufferte Reports (Unbuffered Report Control Blocks, URCBs). Der TwinCAT Telecontrol Configurator generiert Berichtssteuerblockinstanzen die von [FB_AcsiUnbufferedReportControlBlock \[▶ 165\]](#) abgeleitet/erweitert werden. Alle Funktionsbausteine, die von [FB_AcsiUnbufferedReportControlBlock \[▶ 165\]](#) abgeleitet werden, besitzen automatisch eine Default-Implementierung der oben genannten Schnittstelle.

Namensraum: [Tc3_iec61850_8_1 \[▶ 97\]](#)

Bibliothek: [Tc3_iec61850_8_1 \(Tc3_iec61850_8_1.compiled-library\)](#)

Vererbungshierarchie

[I_AcsiCommonNodeClass \[▶ 320\]](#) -> [I_AcsiCommonDataClass \[▶ 311\]](#) -> [I_AcsiCommonControlBlockClass \[▶ 311\]](#) -> [I_AcsiCommonUnbufferedReportControlBlockClass \[▶ 321\]](#) -> [I_ScsmUrCBImplClass](#)

```
INTERFACE I_ScsmUrCBImplClass EXTENDS I_AcsiCommonUnbufferedReportControlBlockClass
```



Eigenschaften

| Name | Type | Zugriff | Beschreibung |
|------------|--|----------|--|
| bGl | BOOL | Get, Set | Attributwert: Aktivierung des Generalabfrage-Befehls. |
| bResv | BOOL | Get, Set | Attributwert: Reservierung des RCB's. |
| bRptEna | BOOL | Get, Set | Attributwert: Aktivierung/Deaktivierung des RCB-Steuerblocks. Weitere Konfiguration benötigt. |
| cOptFields | ST_AcsiOptionalFields [▶ 459] | Get, Set | Attributwert: Konfiguration der optionalen Felder, die mit einem Report übertragen werden. |
| cTrgOps | ST_AcsiTriggerConditions [▶ 466] | Get, Set | Attributwert: Konfiguration der Auslöseoptionen für das Senden eines Reports. |
| nBufTm | DWORD | Get, Set | Attributwert: Konfiguration der max. Report-Pufferzeit in Millisekunden. |
| nConfRev | DWORD | Get, Set | Attributwert: Revisionsnummer der Konfiguration des RCB-Steuerblocks. |
| nIntgPd | DWORD | Get, Set | Attributwert: Konfiguration der max. Zeit zwischen den Integrity-Reports in Millisekunden. |
| nSqNum | BYTE | Get, Set | Attributwert: Report-Sequenznummer. |
| oOwner | T_OCTET64 [▶ 479] | Get, Set | Attributwert: Eigentümer des RCB's. IP-Adresse des Clients welcher den RCB für sich reserviert hat (Offline-/Online-Reservierung). |
| sDatSet | T_AcsiObjectReference [▶ 476] | Get, Set | Attributwert: Name des verknüpften Data-Sets. |
| sRptID | T_AcsiVisString129 [▶ 476] | Get, Set | Attributwert: Report-Identifikationsstring. |
| ipDataSet | I_AcsiCommonDataSetClass [▶ 313] | Get | Schnittstellenzeiger des verknüpften Data-Sets. |
| ipClient | I_ScsmUrCBClientJobClass [▶ 348] | Get | Schnittstellenzeiger der Client-Implementierung des URCB-Steuerblocks. |
| ipServer | I_ScsmUrCBServerJobClass [▶ 349] | Get | Schnittstellenzeiger der Server-Implementierung des URCB-Steuerblocks. |

6.5.3.2.8 I_ScsmUrCBClientJobClass

Objekte, die diese Schnittstelle implementieren, steuern das Verhalten des Client-URCB Steuerblocks.

Namensraum: [Tc3_iec61850_8_1 \[▶ 97\]](#)

Bibliothek: [Tc3_iec61850_8_1 \(Tc3_iec61850_8_1.compiled-library\)](#)

Methoden

| Name | Beschreibung |
|------------------------------------|--------------------------------------|
| EnableReq [▶ 349] | Aktiviert den Steuerblock. |
| DisableReq [▶ 349] | Deaktiviert den Steuerblock. |
| Greq [▶ 349] | Aktiviert den Generalabfrage-Befehl. |

6.5.3.2.8.1 EnableReq

Aktiviert den URCB Steuerblock.

```
METHOD FINAL EnableReq : BOOL
VAR_INPUT
    ipClient : I_ScsmClientClass;
    ipDataSet : I_AcsiCommonDataSetClass;
END_VAR
VAR_OUTPUT
    ipResult : I_AsyncServiceResultClass;
END_VAR
```

ipClient: Schnittstellenzeiger vom Typ: [I_ScsmClientClass \[▶ 351\]](#). Dieser Zeiger bestimmt die Dienstausführende Objektinstanz des IEC 61850 Clients.

ipDataSet: Schnittstellenzeiger vom Typ [I_AcsiCommonDataSetClass \[▶ 313\]](#).

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass \[▶ 387\]](#). Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstausführung überwacht und abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.2.8.2 DisableReq

Deaktiviert den URCB Steuerblock.

```
METHOD FINAL DisableReq : BOOL
VAR_OUTPUT
    ipResult : I_AsyncServiceResultClass;
END_VAR
```

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass \[▶ 387\]](#). Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstausführung überwacht und abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.2.8.3 Greq

Aktiviert den Generalabfrage-Befehl.

```
METHOD FINAL Greq : BOOL
VAR_OUTPUT
    ipResult : I_AsyncServiceResultClass;
END_VAR
```

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass \[▶ 387\]](#). Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstausführung überwacht und abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.2.9 I_ScsmUrCBServerJobClass

Objekte, die diese Schnittstelle implementieren, steuern das Verhalten des Server-URCB Steuerblocks.

Namensraum: [Tc3_iec61850_8_1](#) [[▶ 97](#)]

Bibliothek: Tc3_iec61850_8_1 (Tc3_iec61850_8_1.compiled-library)

Methoden

| Name | Beschreibung |
|---|---|
| Execute [▶ 350] | Führt die RCB-Zustandsmaschine des Servers aus. |

6.5.3.2.9.1 Execute

Führt die RCB-Zustandsmaschine des Servers aus.

METHOD Execute : BOOL

6.5.3.3 I_ScsmAbortIndEventSink

Objekte, die diese Schnittstelle implementieren, besitzen eine benutzerdefinierte Abort-Dienst Ereignisbehandlungsroutine für den Verbindungsabbau.

Namensraum: [Tc3_iec61850_8_1](#) [[▶ 97](#)]

Bibliothek: Tc3_iec61850_8_1 (Tc3_iec61850_8_1.compiled-library)

Vererbungshierarchie

I_ScsmAbortIndEventSink

INTERFACE I_ScsmAbortIndEventSink

Methoden zur Ereignisbehandlung

| Name | Beschreibung |
|--|--|
| OnAbortInd [▶ 350] | Benutzerdefinierte Abort-Dienst Ereignisbehandlungsroutine |

6.5.3.3.1 OnAbortInd

METHOD OnAbortInd

```
VAR_INPUT
    ipAA      : I_ScsmAssociationClass;
    bPAbort   : BOOL;
    ipReason  : I_ServiceErrorClass;
    ipData    : I_BaseStreamClass;
END_VAR
```

ipAA: Schnittstellenzeiger vom Typ: [I_ScsmAssociationClass](#) [[▶ 351](#)]. Über diesen Zeiger kann bei mehreren aktiven Verbindungsinstanzen die Dienstausführende Objektinstanz ermittelt werden. Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

bPAbort: Auslöser der Abort-Indication-Dienstprimitiven. Bei TRUE ausgelöst durch den Dienst-Provider und bei FALSE ausgelöst durch den Dienst-Benutzer (Applikation hat die Abort-Request-Dienstprimitive gestartet).

ipReason: Schnittstellenzeiger vom Typ [I_ServiceErrorClass](#) [[▶ 428](#)]. Über diesen Zeiger können genauere Informationen über die Ursache des Verbindungsabbruchs ermittelt werden.

ipData: Schnittstellenzeiger vom Typ [I_BaseStreamClass](#) [[▶ 395](#)]. Über diesen Zeiger kann auf optionale Daten zugegriffen werden. Beim Zeigerwert = 0 sind keine Daten verfügbar.

6.5.3.4 I_ScsmAssociateCnfEventSink

Objekte, die diese Schnittstelle implementieren, besitzen eine benutzerdefinierte Associate-Dienst Ereignisbehandlungsroutine für den Verbindungsaufbau.

Namensraum: [Tc3_iec61850_8_1 \[▸ 97\]](#)

Bibliothek: Tc3_iec61850_8_1 (Tc3_iec61850_8_1.compiled-library)

Vererbungshierarchie

I_ScsmAssociateCnfEventSink

INTERFACE I_ScsmAssociateCnfEventSink

 **Methoden zur Ereignisbehandlung**

| Name | Beschreibung |
|--|--|
| OnAssociateCnf [▸ 351] | Benutzerdefinierte Associate-Dienst Ereignisbehandlungsroutine |

6.5.3.4.1 OnAssociateCnf

```
METHOD OnAssociateCnf
VAR_INPUT
    ipAA : I_ScsmAssociationClass;
    eError : E_AcsiServiceError;
END_VAR
```

ipAA: Schnittstellenzeiger vom Typ: [I_ScsmAssociationClass \[▸ 351\]](#). Über diesen Zeiger kann bei mehreren aktiven Verbindungsinstanzen die Dienstausführende Objektinstanz ermittelt werden. Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

eError: Aufzählungstyp: [E_AcsiServiceError \[▸ 428\]](#).

6.5.3.5 I_ScsmAssociationClass

Schnittstellenzeiger von diesem Typ identifiziert eine Client-Server-Verbindung.

Namensraum: [Tc3_iec61850_8_1 \[▸ 97\]](#)

Bibliothek: Tc3_iec61850_8_1 (Tc3_iec61850_8_1.compiled-library)

Vererbungshierarchie

I_ScsmAssociationClass

INTERFACE I_ScsmAssociationClass

Methoden

-

Eigenschaften

-

6.5.3.6 I_ScsmClientClass

Objekte, die diese Schnittstelle implementieren steuern den Kommunikationsaufbau und Datenaustausch zwischen der TwinCAT SPS Client-Applikation und dem IEC 61850 server.

Namensraum: [Tc3_iec61850_8_1 \[▸ 97\]](#)

Bibliothek: Tc3_iec61850_8_1 (Tc3_iec61850_8_1.compiled-library)

Vererbungshierarchie

I_ScsmClientClass

INTERFACE I_ScsmClientClass

 Methoden

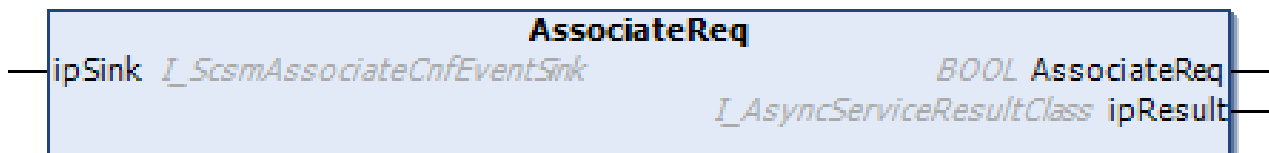
| Name | Beschreibung |
|--|--|
| AssociateReq [▶ 355] | Aktiviert den Dienst: Associate |
| ReleaseReq [▶ 356] | Aktiviert den Dienst: Release |
| ReleaseRsp [▶ 356] | Beantwortet den Dienst: Release |
| AbortReq [▶ 357] | Aktiviert den Dienst: Abort |
| ControlCancelReq [▶ 357] | Aktiviert den Dienst: Cancel |
| ControlOperateReq [▶ 358] | Aktiviert den Dienst: Operate |
| ControlSelectReq [▶ 359] | Aktiviert den Dienst: Select |
| ControlSelectWithValueReq [▶ 360] | Aktiviert den Dienst: SelectWithValue |
| CreateDataSetReq [▶ 361] | Aktiviert den Dienst: CreateDataSet |
| DeleteDataSetReq [▶ 362] | Aktiviert den Dienst: DeleteDataSet |
| GetAllDataValuesReq [▶ 363] | Aktiviert den Dienst: GetAllDataValues |
| GetAllLogicalDeviceValuesReq [▶ 363] | Aktiviert den Dienst: GetAllLogicalDeviceValues (TwinCAT-Spezifisch) |
| GetAllServerValuesReq [▶ 364] | Aktiviert den Dienst: GetAllServerValues (TwinCAT-Spezifisch) |
| GetBrCBValuesReq [▶ 364] | Aktiviert den Dienst: GetBRCBValues |
| GetDataSetDirectoryReq [▶ 365] | Aktiviert den Dienst: GetDataSetDierectory |
| GetDataSetValuesReq [▶ 366] | Aktiviert den Dienst: GetDataSetValues |
| GetDataValuesReq [▶ 366] | Aktiviert den Dienst: GetDataValues |
| GetGoCBValuesReq [▶ 367] | Aktiviert den Dienst: GetGOCBValues |
| GetLogicalDeviceDirectoryReq [▶ 368] | Aktiviert den Dienst: GetLogicalDeviceDirectory |
| GetLogicalNodeDirectoryReq [▶ 368] | Aktiviert den Dienst: GetLogicalNodeDirectory |
| GetServerDirectoryReq [▶ 369] | Aktiviert den Dienst: GetServerDirectory |
| GetUrCBValuesReq [▶ 370] | Aktiviert den Dienst: GetURCBValues |
| SetAllDataValuesReq [▶ 370] | Aktiviert den Dienst: SetAllDataValues (TwinCAT-Spezifisch) |
| SetBrCBValuesReq [▶ 371] | Aktiviert den Dienst: SetBRCBValues |
| SetDataSetValuesReq [▶ 371] | Aktiviert den Dienst: SetDataSetValues (TwinCAT-Spezifisch) |
| SetDataValuesReq [▶ 372] | Aktiviert den Dienst: SetDataValues |
| SetGoCBValuesReq [▶ 373] | Aktiviert den Dienst: SetGOCBValues |
| SetUrCBValuesReq [▶ 373] | Aktiviert den Dienst: SetURCBValues |
| IdentifyReq [▶ 374] | Aktiviert den Dienst: MMS-Identify |
| StatusReq [▶ 375] | Aktiviert den Dienst: MMS-Status |

 **Eigenschaften**

| Name | Typ | Zugriff | Beschreibung |
|---|--|---------|--|
| ipIED [▶ 375] | I_AcsiCommonIntelligentElectronicDeviceClass [▶ 316] | Get | Schnittstellenzeiger von einem Objekt mit der Implementierung der obersten Ebene des IEC 61850 Datenmodells |
| eState | E_AsyncEnvironmentState [▶ 443] | Get | Umgebungszustand der Kommunikationsverbindung. |
| ipSettings | I_ScsmClientSettingsClass [▶ 328] | Get | Protokolleinstellungen der Kommunikationsschicht. |
| ipAssociateCnf | I_ScsmAssociateCnfEventSink [▶ 350] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnAssociateCnf [▶ 351] |
| ipReleaseCnf | I_ScsmReleaseCnfEventSink [▶ 383] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnReleaseCnf [▶ 383]. |
| ipReleaseInd | I_ScsmReleaseIndEventSink [▶ 383] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnReleaseInd [▶ 384]. |
| ipAbortInd | I_ScsmAbortIndEventSink [▶ 350] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnAbortInd [▶ 350] |
| ipSystemClock | I_ScsmSystemClockEventSink [▶ 386] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnGetSystemTime [▶ 387] |
| ipControlCnf | I_ScsmControlCnfEventSink [▶ 375] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnControlCnf [▶ 376] |
| ipCreateDataSetCnf | I_ScsmCreateDataSetCnfEventSink [▶ 377] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnCreateDataSetCnf [▶ 377] |
| ipDeleteDataSetCnf | I_ScsmDeleteDataSetCnfEventSink [▶ 377] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnDeleteDataSetCnf [▶ 378] |
| ipGetDataSetDirectory | I_ScsmGetDataSetDirectoryEventSink [▶ 378] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnGetDataSetDirectoryCnf [▶ 378] |
| ipGetDataValuesCnf | I_ScsmGetDataValuesCnfEventSink [▶ 379] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnGetDataValuesCnf [▶ 379] |
| ipGetLogicalDeviceDirectoryCnf | I_ScsmGetLogicalDeviceDirectoryCnfEventSink [▶ 379] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnGetLogicalDeviceDirectoryCnf [▶ 380] |

| Name | Typ | Zugriff | Beschreibung |
|------------------------------|---|---------|--|
| ipGetLogicalNodeDirectoryCnf | I_ScsmGetLogicalNodeDirectoryCnfEventSink [▶ 380] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnGetLogicalNodeDirectoryCnf [▶ 380] |
| ipGetServerDirectoryCnf | I_ScsmGetServerDirectoryCnfEventSink [▶ 381] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnGetServerDirectoryCnf [▶ 381]. |
| ipLastAppLErrorInd | I_ScsmLastAppLErrorIndEventSink [▶ 382] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnLastAppLErrorInd [▶ 383] |
| ipReportInd | I_ScsmReportIndEventSink [▶ 384] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutinen: OnBufferedReportInd [▶ 384] und OnUnbufferedReportInd [▶ 385] |
| ipSetDataValuesCnf | I_ScsmSetDataValuesCnfEventSink [▶ 385] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnSetDataValuesCnf [▶ 385] |
| ipIdentifyCnf | I_ScsmIdentifyCnfEventSink [▶ 382] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnIdentifyCnf [▶ 382] |
| ipStatusCnf | I_ScsmStatusCnfEventSink [▶ 386] | Set | Schnittstellenzeiger von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnStatusCnf [▶ 386] |

6.5.3.6.1 AssociateReq



Diese Methode aktiviert den Dienst Associate. Mit diesem Dienst kann ein Client eine Verbindung zum Server aufbauen.

```

METHOD AssociateReq : BOOL
VAR_INPUT
    ipSink : I_ScsmAssociateCnfEventSink;
END_VAR
VAR_OUTPUT
    ipResult : I_AsyncServiceResultClass;
END_VAR
    
```

ipSink: Schnittstellenzeiger vom Typ [I_ScsmAssociateCnfEventSink](#) [▶ 350] von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine [OnAssociateCnf](#) (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass](#) [▶ 387]. Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstausführung überwacht und abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

Beispiel (Auszug):**Deklarationsteil:**

```

FUNCTION_BLOCK FB_My61850Client IMPLEMENTS I_ScsmAbortEventSink
VAR
    fbClient      : FB_IEC61850CommonClass := ( ipIED := IED, settings := ( nRemotePort := 102, s
RemoteHost := '192.168.10.145' ), ipAbort := THIS^ );
    state         : BYTE;
    bSuccess      : BOOL;
    ipAsyncResult : I_AsyncServiceResultClass;
END_VAR

```

Implementierung:

```

CASE state OF
    0:(* idle state *)
        IF bAssociateReq THEN(* Establish connection *)
            bAssociateReq := FALSE;
            bSuccess := fbClient.AssociateReq( ipSink := 0, ipResult=>ipAsyncResult );
            state := SEL( bSuccess, 100, 1 );
        END_IF
    1:(* wait until connection established *)
        IF ipAsyncResult <> 0 THEN
            IF NOT ipAsyncResult.IsBusy() THEN
                state := SEL( ipAsyncResult.IsCompleted(), 100, 10 );
            END_IF
        END_IF
    10:(* connection established *)
        ;
    100:(* error state *)
        state := 0;
END_CASE

```

6.5.3.6.2 ReleaseReq

Diese Methode aktiviert den Dienst Release. Mit diesem Dienst kann ein Client eine Verbindung zum Server geregelt abbauen.

```

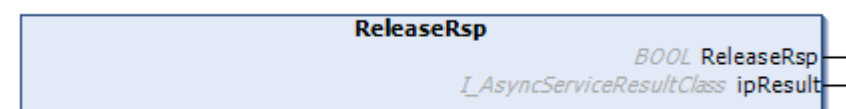
METHOD ReleaseReq : BOOL
VAR_INPUT
    ipSink : I_ScsmReleaseCnfEventSink;
END_VAR
VAR_OUTPUT
    ipResult : I_AsyncServiceResultClass;
END_VAR

```

ipSink: Schnittstellenzeiger vom Typ [I_ScsmReleaseCnfEventSink \[► 383\]](#) von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine OnReleaseCnf (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass \[► 387\]](#). Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.3 ReleaseRsp

Diese Methode beantwortet den Dienst Release. Mit diesem Dienst kann ein Client eine Verbindung zum Server geregelt abbauen.

```
METHOD ReleaseRsp : BOOL
VAR_OUTPUT
    ipResult : I_AsyncServiceResultClass;
END_VAR
```

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass \[▶ 387\]](#). Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.4 AbortReq



Diese Methode aktiviert den Dienst Abort. Mit diesem Dienst kann ein Client die Verbindung zum Server ungeregelt abbauen.

```
METHOD AbortReq : BOOL
VAR_INPUT
    ipReason : I_ServiceErrorClass;
    ipSink : I_ScsmAbortIndEventSink;
END_VAR
VAR_OUTPUT
    ipResult : I_AsyncServiceResultClass;
END_VAR
```

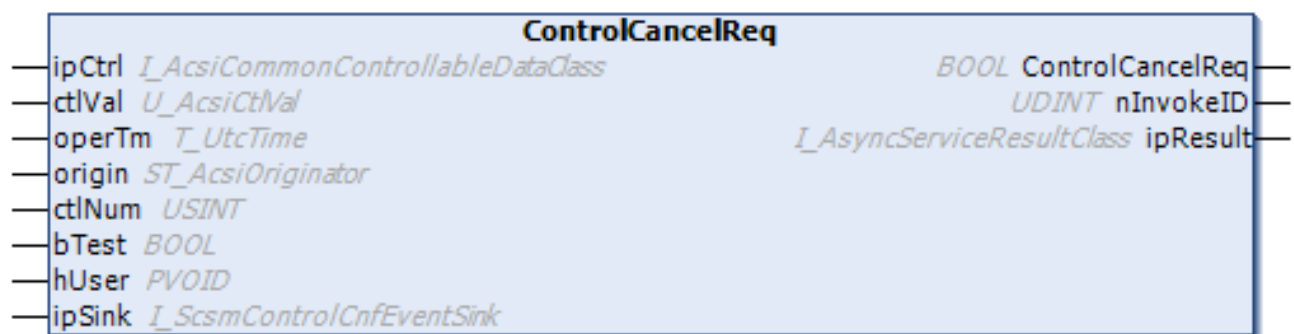
ipReason: Schnittstellenzeiger vom Typ [I_ServiceErrorClass \[▶ 392\]](#). Über diesen Zeiger kann ein Objekt mit einer benutzerdefinierten Ursache für den Verbindungsabbruch übergeben werden.

ipSink: Schnittstellenzeiger vom Typ [I_ScsmAbortIndEventSink \[▶ 350\]](#) von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnAbortInd (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass \[▶ 387\]](#). Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.5 ControlCancelReq



Diese Methode aktiviert den Dienst Cancel.

```
METHOD ControlCancelReq : BOOL
VAR_INPUT
    ipCtrl : I_AcsiCommonControllableDataClass;
```

```

    ctlVal      : U_AcsiCtlVal;
    operTm     : T_UtcTime;
    origin     : ST_AcsiOriginator;
    ctlNum     : USINT;
    bTest      : BOOL;
    hUser      : PVOID;
    ipSink     : I_ScsmControlCnfEventSink;
END_VAR
VAR_OUTPUT
    nInvokeID : UDINT;
    ipResult  : I_AsyncServiceResultClass;
END_VAR

```

ipCtrl: Schnittstellenzeiger vom Typ [I_AcsiCommonControllableDataClass](#) [▶ 311] von einem steuerbaren Objekt. Auf Objekten, die diese Schnittstelle implementieren, können Dienste für die Schaltersteuerung wie Operate, Select, SelectWithValue oder Cancel ausgeführt werden. Zu solchen Objekten gehören z. B. Instanzen vom „Steuerpunkt mit Doppelmeldung“ (Controllable Double Point, DPC) oder „Steuerpunkt mit Einfachmeldung“ (Controllable Single Point, SPC).

ctlVal: Parameter für den neuen Steuerwert (control value) vom Typ [U_AcsiCtlVal](#) [▶ 481].

operTm: Parameter für die Schaltzeit (operation time) vom Typ [T_UtcTime](#) [▶ 479] für eine zeitgesteuerte Befehlsausführung. Setzen Sie diesen Parameter auf den Wert: `operTm := AcsiConstants.NULL_TimeStamp` wenn der steuerbare Objekt den Dienst: `TimeActivatedOperate` nicht unterstützt.

origin: Parameter für den Befehlsursprung vom Typ [ST_AcsiOriginator](#) [▶ 460] über den ein Client der den Befehl absetzt identifiziert werden kann.

ctlNum: Kennung (z.B.: Fortlaufende Nummer) für den Befehl über den ein Client die empfangenen negativen oder positiven Befehlsbestätigungen zum Request zuordnen kann.

bTest: Test Flag.

hUser: Optionaler benutzerdefinierter Parameter oder auch Zeiger. Ein beliebiger Wert für diesen Parameter kann in der Methoden für Dienstaktivierung (`XyzMethodReq`) übergeben werden. In der Ereignisbehandlungsroutine der Dienstrückmeldung wird dann der gleiche Wert gespiegelt.

ipSink: Schnittstellenzeiger vom Typ [I_ScsmControlCnfEventSink](#) [▶ 375] von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: `OnControlCnf` (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass](#) [▶ 387]. Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

nInvokeID: Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (`XyzMethodReq`) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (`OnXyzMethodCnf`).

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6 ControlOperateReq



Diese Methode aktiviert den Dienst Operate.

```
METHOD ControlOperateReq : BOOL
VAR_INPUT
    ipCtrl      : I_AcsiCommonControllableDataClass;
    ctlVal      : U_AcsiCtlVal;
    operTm      : T_UtcTime;
    origin      : ST_AcsiOriginator;
    ctlNum      : USINT;
    bTest       : BOOL;
    eCheck      : E_AcsiCheck;
    hUser       : PVOID;
    ipSink      : I_ScsmControlCnfEventSink;
END_VAR
VAR_OUTPUT
    nInvokeID   : UDINT;
    ipResult    : I_AsyncServiceResultClass;
END_VAR
```

ipCtrl: Schnittstellenzeiger vom Typ [I_AcsiCommonControllableDataClass](#) [▶ 311] von einem steuerbaren Objekt. Auf Objekten, die diese Schnittstelle implementieren, können Dienste für die Schaltersteuerung wie Operate, Select, SelectWithValue oder Cancel ausgeführt werden. Zu solchen Objekten gehören z. B. Instanzen vom „Steuerpunkt mit Doppelmeldung“ (Controllable Double Point, DPC) oder „Steuerpunkt mit Einfachmeldung“ (Controllable Single Point, SPC).

ctlVal: Parameter für den neuen Steuerwert (control value) vom Typ [U_AcsiCtlVal](#) [▶ 481].

operTm: Parameter für die Schaltzeit (operation time) vom Typ [T_UtcTime](#) [▶ 479] für eine zeitgesteuerte Befehlsausführung. Setzen Sie diesen Parameter auf den Wert: operTm := AcsiConstants.NULL_TimeStamp wenn der steuerbare Objekt den Dienst: TimeActivatedOperate nicht unterstützt.

origin: Parameter für den Befehlsursprung vom Typ [ST_AcsiOriginator](#) [▶ 460] über den ein Client der den Befehl absetzt identifiziert werden kann.

ctlNum: Kennung (z.B.: Fortlaufende Nummer) für den Befehl über den ein Client die empfangenen negativen oder positiven Befehlsbestätigungen zum Request zuordnen kann.

bTest: Test Flag.

eCheck: Parameter vom Typ [E_AcsiCheck](#) [▶ 401] für Synchro-Check und/oder Interlocking-Tests.

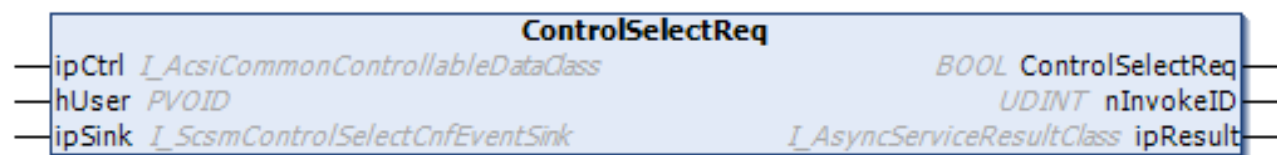
ipSink: Schnittstellenzeiger vom Typ [I_ScsmControlCnfEventSink](#) [▶ 375] von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnControlCnf (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass](#) [▶ 387]. Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

nInvokeID: Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (XyzMethodReq) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (OnXyzMethodCnf).

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.7 ControlSelectReq



Diese Methode aktiviert den Dienst Select.

```
METHOD ControlSelectReq : BOOL
VAR_INPUT
    ipCtrl      : I_AcsiCommonControllableDataClass;
    hUser       : PVOID;
END_VAR
```

```

    ipSink      : I_ScsmControlCnfEventSink;
END_VAR
VAR_OUTPUT
    nInvokeID  : UDINT;
    ipResult   : I_AsyncServiceResultClass;
END_VAR

```

ipCtrl: Schnittstellenzeiger vom Typ [I_AcsiCommonControllableDataClass](#) [▶ 311] von einem steuerbaren Objekt. Auf Objekten, die diese Schnittstelle implementieren, können Dienste für die Schaltersteuerung wie Operate, Select, SelectWithValue oder Cancel ausgeführt werden. Zu solchen Objekten gehören z. B. Instanzen vom „Steuerpunkt mit Doppelmeldung“ (Controllable Double Point, DPC) oder „Steuerpunkt mit Einfachmeldung“ (Controllable Single Point, SPC).

hUser: Optionaler benutzerdefinierter Parameter oder auch Zeiger. Ein beliebiger Wert für diesen Parameter kann in der Methoden für Dienstaktivierung (XyzMethodReq) übergeben werden. In der Ereignisbehandlungsroutine der Dienstrückmeldung wird dann der gleiche Wert gespiegelt.

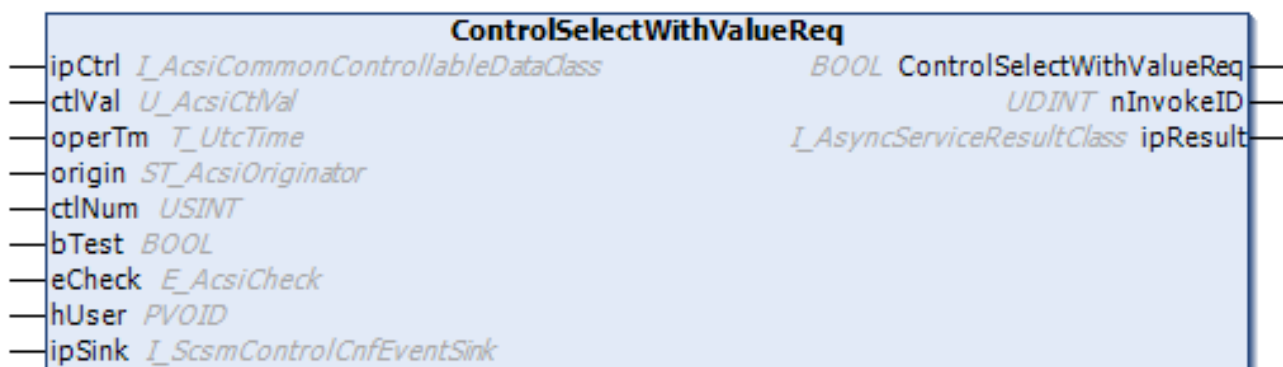
ipSink: Schnittstellenzeiger vom Typ [I_ScsmControlCnfEventSink](#) [▶ 375] von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnControlCnf (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass](#) [▶ 387]. Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

nInvokeID: Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (XyzMethodReq) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (OnXyzMethodCnf).

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.8 ControlSelectWithValueReq



Diese Methode aktiviert den Dienst SelectWithValue.

```

METHOD ControlSelectWithValueReq : BOOL
VAR_INPUT
    ipCtrl      : I_AcsiCommonControllableDataClass;
    ctlVal      : U_AcsiCtlVal;
    operTm      : T_UtcTime;
    origin      : ST_AcsiOriginator;
    ctlNum      : USINT;
    bTest       : BOOL;
    eCheck      : E_AcsiCheck;
    hUser       : PVOID;
    ipSink      : I_ScsmControlCnfEventSink;
END_VAR
VAR_OUTPUT
    nInvokeID   : UDINT;
    ipResult    : I_AsyncServiceResultClass;
END_VAR

```

ipCtrl: Schnittstellenzeiger vom Typ [I_AcsiCommonControllableDataClass](#) [▶ 311] von einem steuerbaren Objekt. Auf Objekten, die diese Schnittstelle implementieren, können Dienste für die Schaltersteuerung wie Operate, Select, SelectWithValue oder Cancel ausgeführt werden. Zu solchen Objekten gehören z. B. Instanzen vom „Steuerpunkt mit Doppelmeldung“ (Controllable Double Point, DPC) oder „Steuerpunkt mit Einfachmeldung“ (Controllable Single Point, SPC).

ctlVal: Parameter für den neuen Steuerwert (control value) vom Typ [U_AcsiCtlVal](#) [▶ 481].

operTm: Parameter für die Schaltzeit (operation time) vom Typ [T_UtcTime](#) [▶ 479] für eine zeitgesteuerte Befehlsausführung. Setzen Sie diesen Parameter auf den Wert: `operTm := AcsiConstants.NULL_TimeStamp` wenn der steuerbare Objekt den Dienst: `TimeActivatedOperate` nicht unterstützt.

origin: Parameter für den Befehlsursprung vom Typ [ST_AcsiOriginator](#) [▶ 460] über den ein Client der den Befehl absetzt identifiziert werden kann.

ctlNum: Kennung (z.B.: Fortlaufende Nummer) für den Befehl über den ein Client die empfangenen negativen oder positiven Befehlsbestätigungen zum Request zuordnen kann.

bTest: Test Flag.

eCheck: Parameter vom Typ [E_AcsiCheck](#) [▶ 401] für Synchro-Check und/oder Interlocking-Tests.

hUser: Optionaler benutzerdefinierter Parameter oder auch Zeiger. Ein beliebiger Wert für diesen Parameter kann in der Methoden für Dienstaktivierung (`XYZMethodReq`) übergeben werden. In der Ereignisbehandlungsroutine der Dienstrückmeldung wird dann der gleiche Wert gespiegelt.

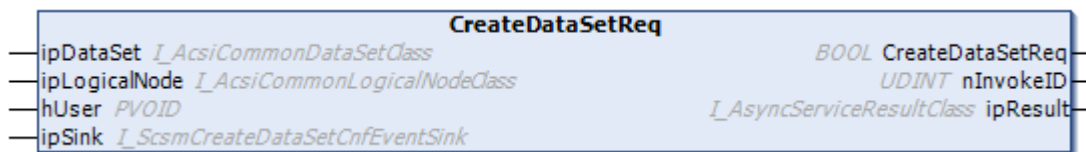
ipSink: Schnittstellenzeiger vom Typ [I_ScsmControlCnfEventSink](#) [▶ 375] von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: `OnControlCnf` (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass](#) [▶ 387]. Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

nInvokeID: Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (`XYZMethodReq`) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (`OnXYZMethodCnf`).

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.9 CreateDataSetReq



Diese Methode aktiviert den Dienst: `CreateDataSet`. Mit diesem Dienst kann der Client dynamisch (zur Laufzeit) einen „persistent“ oder „non-persistent“ Datensatz (Data-Set) erstellen.

```
METHOD CreateDataSetReq : BOOL
VAR_INPUT
    ipDataSet      : I_AcsiCommonDataSetClass;
    ipLogicalNode  : I_AcsiCommonLogicalNodeClass;
    hUser          : PVOID;
    ipSink         : I_ScsmCreateDataSetCnfEventSink;
END_VAR
VAR_OUTPUT
    nInvokeID     : UDINT;
    ipResult      : I_AsyncServiceResultClass;
END_VAR
```

ipDataSet: Schnittstellenzeiger vom Typ [I_AcsiCommonDataSetClass](#) [▶ 313].

ipLogicalNode: Schnittstellenzeiger vom Typ [I_AcsiCommonLogicalNodeClass](#) [▶ 317].

Der Zeiger auf den logischen Knoten bestimmt, ob ein „persistent“ oder „non-persistent“ Datensatz erstellt werden soll. Die „persistent“ Datensätze werden mit einem logischen Knoten verknüpft. In diesem Fall muss dieser Parameter gültig sein. Bei einem „non-persistent“ Datensatz hat der Parameter den Wert Null.

hUser: Optionaler benutzerdefinierter Parameter oder auch Zeiger. Ein beliebiger Wert für diesen Parameter kann in der Methoden für Dienstaktivierung (XyzMethodReq) übergeben werden. In der Ereignisbehandlungsroutine der Dienstrückmeldung wird dann der gleiche Wert gespiegelt.

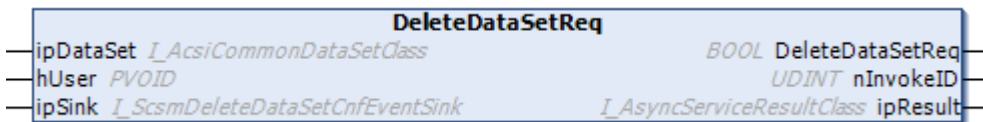
ipSink: Schnittstellenzeiger vom Typ: [I_ScsmCreateDataSetCnfEventSink \[▶ 377\]](#) von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnCreateDataSetCnf (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

nInvokeID: Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (XyzMethodReq) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (OnXyzMethodCnf).

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass \[▶ 387\]](#). Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.10 DeleteDataSetReq



Diese Methode aktiviert den Dienst: DeleteDataSet. Mit diesem Dienst kann der Client eine dynamisch (zur Laufzeit) erstellte Datensatzinstanz (Data-Set) löschen.

```
METHOD DeleteDataSetReq : BOOL
VAR_INPUT
    ipDataSet : I_AcsiCommonDataSetClass;
    hUser     : PVOID;
    ipSink    : I_ScsmDeleteDataSetCnfEventSink;
END_VAR
VAR_OUTPUT
    nInvokeID : UDINT;
    ipResult  : I_AsyncServiceResultClass;
END_VAR
```

ipDataSet: Schnittstellenzeiger vom Typ [I_AcsiCommonDataSetClass \[▶ 313\]](#).

hUser: Optionaler benutzerdefinierter Parameter oder auch Zeiger. Ein beliebiger Wert für diesen Parameter kann in der Methoden für Dienstaktivierung (XyzMethodReq) übergeben werden. In der Ereignisbehandlungsroutine der Dienstrückmeldung wird dann der gleiche Wert gespiegelt.

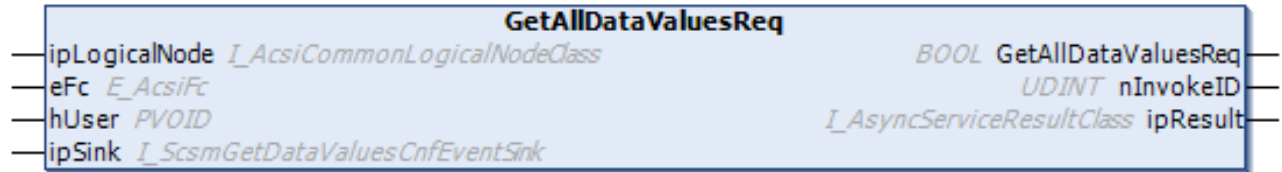
ipSink: Schnittstellenzeiger vom Typ: [I_ScsmDeleteDataSetCnfEventSink \[▶ 377\]](#) von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnDeleteDataSetCnf (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

nInvokeID: Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (XyzMethodReq) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (OnXyzMethodCnf).

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass \[▶ 387\]](#). Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.11 GetAllDataValuesReq



Diese Methode aktiviert den Dienst GetAllDataValues.

```
METHOD GetAllDataValuesReq : BOOL
VAR_INPUT
    ipLogicalNode : I_AcsiCommonLogicalNodeClass;
    eFc           : E_AcsiFc;
    hUser        : PVOID;
    ipSink       : I_ScsmGetDataValuesCnfEventSink;
END_VAR
VAR_OUTPUT
    nInvokeID    : UDINT;
    ipResult     : I_AsyncServiceResultClass;
END_VAR
```

ipLogicalNode: Schnittstellenzeiger vom Typ [I_AcsiCommonLogicalNodeClass](#) [► 317].

eFc: Funktionale Gruppe vom Typ [E_AcsiFc](#) [► 409].

hUser: Optionaler benutzerdefinierter Parameter oder auch Zeiger. Ein beliebiger Wert für diesen Parameter kann in der Methoden für Dienstaktivierung (XyzMethodReq) übergeben werden. In der Ereignisbehandlungsroutine der Dienstrückmeldung wird dann der gleiche Wert gespiegelt.

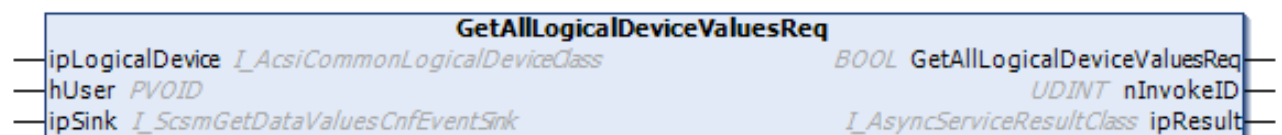
ipSink: Schnittstellenzeiger vom Typ [I_ScsmGetDataValuesCnfEventSink](#) [► 379] von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnGetDataValuesCnf (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass](#) [► 387]. Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

nInvokeID: Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (XyzMethodReq) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (OnXyzMethodCnf).

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.12 GetAllLogicalDeviceValuesReq



Diese Methode aktiviert den Dienst GetAllLogicalDeviceValues.

```
METHOD GetAllLogicalDeviceValuesReq : BOOL
VAR_INPUT
    ipLogicalDevice : I_AcsiCommonLogicalDeviceClass;
    hUser          : PVOID;
    ipSink         : I_ScsmGetDataValuesCnfEventSink;
END_VAR
VAR_OUTPUT
    nInvokeID      : UDINT;
    ipResult       : I_AsyncServiceResultClass;
END_VAR
```

ipLogicalDevice: Schnittstellenzeiger vom Typ [I_AcsiCommonLogicalDeviceClass](#) [► 317].

hUser: Optionaler benutzerdefinierter Parameter oder auch Zeiger. Ein beliebiger Wert für diesen Parameter kann in der Methoden für Dienstaktivierung (XyzMethodReq) übergeben werden. In der Ereignisbehandlungsroutine der Dienstrückmeldung wird dann der gleiche Wert gespiegelt.

ipSink: Schnittstellenzeiger vom Typ [I_ScsmGetDataValuesCnfEventSink \[▶ 379\]](#) von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnGetDataValuesCnf (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass \[▶ 387\]](#). Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

nInvokeID: Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (XyzMethodReq) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (OnXyzMethodCnf).

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.13 GetAllServerValuesReq

| GetAllServerValuesReq | |
|--|------------------------------------|
| ipServer <i>I_AcsiCommonIntelligentElectronicDeviceClass</i> | BOOL GetAllServerValuesReq |
| hUser <i>PVOID</i> | UDINT nInvokeID |
| ipSink <i>I_ScsmGetDataValuesCnfEventSink</i> | I_AsyncServiceResultClass ipResult |

Diese Methode aktiviert den Dienst GetAllServerValues.

```
METHOD GetAllServerValuesReq : BOOL
VAR_INPUT
    ipServer : I_AcsiCommonIntelligentElectronicDeviceClass;
    hUser    : PVOID;
    ipSink   : I_ScsmGetDataValuesCnfEventSink;
END_VAR
VAR_OUTPUT
    nInvokeID : UDINT;
    ipResult  : I_AsyncServiceResultClass;
END_VAR
```

ipServer: Schnittstellenzeiger vom Typ [I_AcsiCommonIntelligentElectronicDeviceClass \[▶ 316\]](#).

hUser: Optionaler benutzerdefinierter Parameter oder auch Zeiger. Ein beliebiger Wert für diesen Parameter kann in der Methoden für Dienstaktivierung (XyzMethodReq) übergeben werden. In der Ereignisbehandlungsroutine der Dienstrückmeldung wird dann der gleiche Wert gespiegelt.

ipSink: Schnittstellenzeiger vom Typ [I_ScsmGetDataValuesCnfEventSink \[▶ 379\]](#) von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnGetDataValuesCnf (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass \[▶ 387\]](#). Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

nInvokeID: Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (XyzMethodReq) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (OnXyzMethodCnf).

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.14 GetBrCBValuesReq

| GetBrCBValuesReq | |
|---|------------------------------------|
| ipBrCb <i>I_AcsiCommonBufferedReportControlBlockClass</i> | BOOL GetBrCBValuesReq |
| hUser <i>PVOID</i> | UDINT nInvokeID |
| ipSink <i>I_ScsmGetDataValuesCnfEventSink</i> | I_AsyncServiceResultClass ipResult |

Diese Methode aktiviert den Dienst GetBrCBValues. Mit diesem Dienst kann der Client die Werte aller Datenobjekte eines Buffered Report Control Blocks auslesen.


```
METHOD GetBRCBValuesReq : BOOL
VAR_INPUT
    ipBrcb      : I_AcsiCommonBufferedReportControlBlockClass;
    hUser       : PVOID;
    ipSink      : I_ScsmGetDataValuesCnfEventSink;
END_VAR
VAR_OUTPUT
    nInvokeID  : UDINT;
    ipResult   : I_AsyncServiceResultClass;
END_VAR
```

ipBrcb: Schnittstellenzeiger vom Typ [I_AcsiCommonBufferedReportControlBlockClass](#) [► 310].

hUser: Optionaler benutzerdefinierter Parameter oder auch Zeiger. Ein beliebiger Wert für diesen Parameter kann in der Methoden für Dienstaktivierung (XyzMethodReq) übergeben werden. In der Ereignisbehandlungsroutine der Dienstrückmeldung wird dann der gleiche Wert gespiegelt.

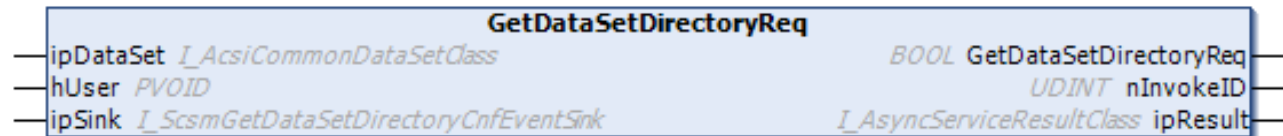
ipSink: Schnittstellenzeiger vom Typ [I_ScsmGetDataValuesCnfEventSink](#) [► 379] von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnGetDataValuesCnf (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass](#) [► 387]. Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

nInvokeID: Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (XyzMethodReq) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (OnXyzMethodCnf).

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.15 GetDataSetDirectoryReq



Diese Methode aktiviert den Dienst GetDataSetDirectory. Mit diesem Dienst kann ein Client beim Server abfragen, welche Datenobjekte in einem bestimmten Data Set referenziert werden.

```
METHOD GetDataSetDirectoryReq : BOOL
VAR_INPUT
    ipDataSet   : I_AcsiCommonDataSetClass;
    hUser       : PVOID;
    ipSink      : I_ScsmGetDataSetDirectoryCnfEventSink;
END_VAR
VAR_OUTPUT
    nInvokeID  : UDINT;
    ipResult   : I_AsyncServiceResultClass;
END_VAR
```

ipDataSet: Schnittstellenzeiger vom Typ [I_AcsiCommonDataSetClass](#) [► 313].

hUser: Optionaler benutzerdefinierter Parameter oder auch Zeiger. Ein beliebiger Wert für diesen Parameter kann in der Methoden für Dienstaktivierung (XyzMethodReq) übergeben werden. In der Ereignisbehandlungsroutine der Dienstrückmeldung wird dann der gleiche Wert gespiegelt.

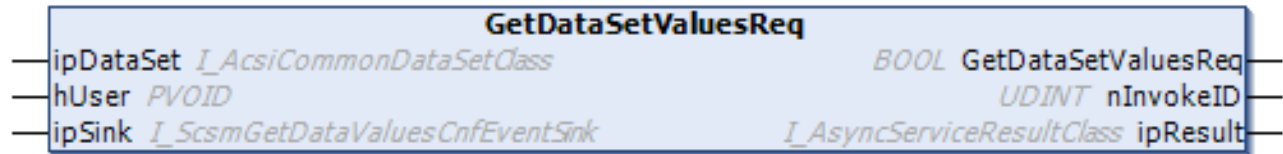
ipSink: Schnittstellenzeiger vom Typ [I_ScsmGetDataSetDirectoryCnfEventSink](#) [► 378] von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnGetDataSetDirectoryCnf (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass](#) [► 387]. Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

nInvokeID: Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (XyzMethodReq) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (OnXyzMethodCnf).

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.16 GetDataSetValuesReq



Diese Methode aktiviert den Dienst GetDataSetValues.

```
METHOD GetDataSetValuesReq : BOOL
VAR_INPUT
  ipDataSet : I_AcsiCommonDataSetClass;
  hUser     : PVOIDID;
  ipSink    : I_ScsmGetDataValuesCnfEventSink;
END_VAR
VAR_OUTPUT
  nInvokeID : UDINT;
  ipResult  : I_AsyncServiceResultClass;
END_VAR
```

ipDataSet: Schnittstellenzeiger vom Typ [I_AcsiCommonDataSetClass](#) [► 313].

hUser: Optionaler benutzerdefinierter Parameter oder auch Zeiger. Ein beliebiger Wert für diesen Parameter kann in der Methoden für Dienstaktivierung (XyzMethodReq) übergeben werden. In der Ereignisbehandlungsroutine der Dienstrückmeldung wird dann der gleiche Wert gespiegelt.

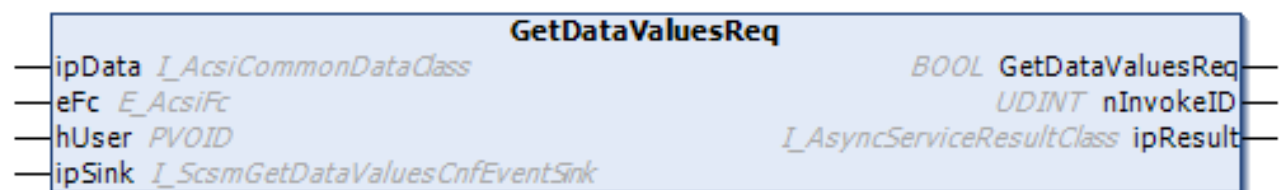
ipSink: Schnittstellenzeiger vom Typ [I_ScsmGetDataValuesCnfEventSink](#) [► 379] von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnGetDataValuesCnf (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass](#) [► 387]. Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

nInvokeID: Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (XyzMethodReq) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (OnXyzMethodCnf).

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.17 GetDataValuesReq



Diese Methode aktiviert den Dienst: GetDataValues.

```
METHOD GetDataValuesReq : BOOL
VAR_INPUT
  ipData : I_AcsiCommonDataClass;
  eFc    : E_AcsiFc;
  hUser  : PVOIDID;
  ipSink : I_ScsmGetDataValuesCnfEventSink;
END_VAR
VAR_OUTPUT
  nInvokeID : UDINT;
  ipResult  : I_AsyncServiceResultClass;
END_VAR
```

ipData: Schnittstellenzeiger vom Typ [I_AcsiCommonDataClass](#) [► 311].

eFc: Funktionale Gruppe vom Typ [E_AcsiFc](#) [[▶ 409](#)].

hUser: Optionaler benutzerdefinierter Parameter oder auch Zeiger. Ein beliebiger Wert für diesen Parameter kann in der Methoden für Dienstaktivierung (XyzMethodReq) übergeben werden. In der Ereignisbehandlungsroutine der Dienstrückmeldung wird dann der gleiche Wert gespiegelt.

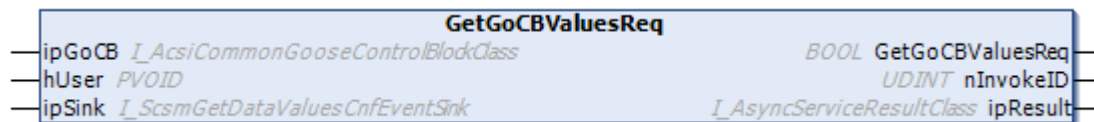
ipSink: Schnittstellenzeiger vom Typ [I_ScsmGetDataValuesCnfEventSink](#) [[▶ 379](#)] von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnGetDataValuesCnf (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass](#) [[▶ 387](#)]. Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

nInvokeID: Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (XyzMethodReq) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (OnXyzMethodCnf).

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.18 GetGoCBValuesReq



Diese Methode aktiviert den Dienst GetGOCBValues. Mit diesem Dienst kann der Client die Werte aller Datenobjekte eines GOOSE-Steuerblocks (GOOSE Control Block) auslesen.

```

METHOD GetGoCBValuesReq : BOOL
VAR_INPUT
    ipGoCB      : I_AcsiCommonGooseControlBlockClass;
    hUser       : PVOIDID;
    ipSink      : I_ScsmGetDataValuesCnfEventSink;
END_VAR
VAR_OUTPUT
    nInvokeID   : UDINT;
    ipResult    : I_AsyncServiceResultClass;
END_VAR
    
```

ipGoCB: Schnittstellenzeiger vom Typ [I_AcsiCommonGooseControlBlockClass](#) [[▶ 315](#)].

hUser: Optionaler benutzerdefinierter Parameter oder auch Zeiger. Ein beliebiger Wert für diesen Parameter kann in der Methoden für Dienstaktivierung (XyzMethodReq) übergeben werden. In der Ereignisbehandlungsroutine der Dienstrückmeldung wird dann der gleiche Wert gespiegelt.

ipSink: Schnittstellenzeiger vom Typ [I_ScsmGetDataValuesCnfEventSink](#) [[▶ 379](#)] von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnGetDataValuesCnf (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

nInvokeID: Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (XyzMethodReq) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (OnXyzMethodCnf).

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass](#) [[▶ 387](#)]. Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.19 GetLogicalDeviceDirectoryReq



Diese Methode aktiviert den Dienst GetLogicalDeviceDirectory. Mit diesem Dienst kann ein Client beim Server abfragen, welche Logische Knoten in einem bestimmten Logischen Gerät implementiert werden.

```
METHOD GetLogicalDeviceDirectoryReq : BOOL
VAR_INPUT
    ipLogicalDevice : I_AcsiCommonLogicalDeviceClass;
    hUser           : PVOID;
    ipSink          : I_ScsmGetLogicalDeviceDirectoryCnfEventSink;
END_VAR
VAR_OUTPUT
    nInvokeID      : UDINT;
    ipResult       : I_AsyncServiceResultClass;
END_VAR
```

ipLogicalDevice: Schnittstellenzeiger vom Typ [I_AcsiCommonLogicalDeviceClass](#) [▶ 317].

hUser: Optionaler benutzerdefinierter Parameter oder auch Zeiger. Ein beliebiger Wert für diesen Parameter kann in der Methoden für Dienstaktivierung (XyzMethodReq) übergeben werden. In der Ereignisbehandlungsroutine der Dienstrückmeldung wird dann der gleiche Wert gespiegelt.

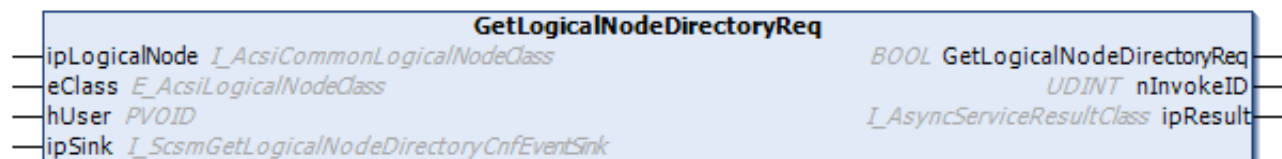
ipSink: Schnittstellenzeiger vom Typ [I_ScsmGetLogicalDeviceDirectoryCnfEventSink](#) [▶ 379] von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine OnGetLogicalDeviceDirectoryCnf (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass](#) [▶ 387]. Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

nInvokeID: Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (XyzMethodReq) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (OnXyzMethodCnf).

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.20 GetLogicalNodeDirectoryReq



Diese Methode aktiviert den Dienst GetLogicalNodeDirectory. Mit diesem Dienst kann ein Client beim Server abfragen, welche Datenobjekte, Datengruppen, Control-Blöcke in einem bestimmten Logischen Knoten implementiert werden.

```
METHOD GetLogicalNodeDirectoryReq : BOOL
VAR_INPUT
    ipLogicalNode : I_AcsiCommonLogicalNodeClass;
    eClass        : E_AcsiLogicalNodeClass;
    hUser         : PVOID;
    ipSink        : I_ScsmGetLogicalNodeDirectoryCnfEventSink;
END_VAR
VAR_OUTPUT
    nInvokeID     : UDINT;
    ipResult      : I_AsyncServiceResultClass;
END_VAR
```

ipLogicalNode: Schnittstellenzeiger vom Typ [I_AcsiCommonLogicalNodeClass](#) [▶ 317].

eClass: Objektklasse des Logischen Knoten vom Typ [E_AcsiLogicalNodeClass](#) [▶ 414].

hUser: Optionaler benutzerdefinierter Parameter oder auch Zeiger. Ein beliebiger Wert für diesen Parameter kann in der Methoden für Dienstaktivierung (XyzMethodReq) übergeben werden. In der Ereignisbehandlungsroutine der Dienstrückmeldung wird dann der gleiche Wert gespiegelt.

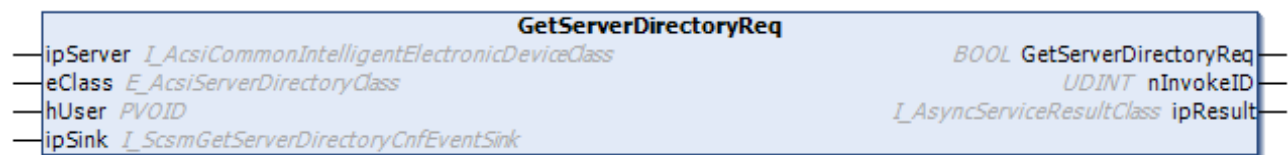
ipSink: Schnittstellenzeiger vom Typ [I_ScsmGetLogicalNodeDirectoryCnfEventSink \[▶ 380\]](#) von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnGetLogicalNodeDirectoryCnf (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass \[▶ 387\]](#). Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

nInvokeID: Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (XyzMethodReq) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (OnXyzMethodCnf).

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.21 GetServerDirectoryReq



Diese Methode aktiviert den Dienst GetServerDirectoryReq. Mit diesem Dienst kann ein Client abfragen, welche Logischen Geräte oder Dateien in einem bestimmten Server implementiert werden.

```

METHOD GetServerDirectoryReq : BOOL
VAR_INPUT
    ipServer : I_AcsiCommonIntelligentElectronicDeviceClass;
    eClass   : E_AcsiServerDirectoryClass;
    hUser    : PVOID;
    ipSink   : I_ScsmGetServerDirectoryCnfEventSink;
END_VAR
VAR_OUTPUT
    nInvokeID : UDINT;
    ipResult  : I_AsyncServiceResultClass;
END_VAR
    
```

ipServer: Schnittstellenzeiger vom Typ [I_AcsiCommonIntelligentElectronicDeviceClass \[▶ 316\]](#).

eClass: Server Klasse vom Typ [E_AcsiServerDirectoryClass \[▶ 428\]](#). Dieser Parameter legt fest, ob Logischen Geräte oder Dateien abgefragt werden sollen.

hUser: Optionaler benutzerdefinierter Parameter oder auch Zeiger. Ein beliebiger Wert für diesen Parameter kann in der Methoden für Dienstaktivierung (XyzMethodReq) übergeben werden. In der Ereignisbehandlungsroutine der Dienstrückmeldung wird dann der gleiche Wert gespiegelt.

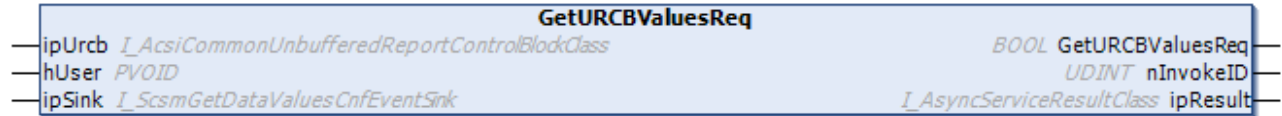
ipSink: Schnittstellenzeiger vom Typ [I_ScsmGetServerDirectoryCnfEventSink \[▶ 381\]](#) von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine OnGetServerDirectoryCnf. Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass \[▶ 387\]](#). Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

nInvokeID: Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (XyzMethodReq) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (OnXyzMethodCnf).

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.22 GetUrCBValuesReq



Diese Methode aktiviert den Dienst GetURCBValues.

```

METHOD GetURCBValuesReq : BOOL
VAR_INPUT
    ipUrcb      : I_AcsiCommonUnbufferedReportControlBlockClass;
    hUser       : PVOID;
    ipSink      : I_ScsmGetDataValuesCnfEventSink;
END_VAR
VAR_OUTPUT
    nInvokeID  : UDINT;
    ipResult   : I_AsyncServiceResultClass;
END_VAR

```

ipUrcb: Schnittstellenzeiger vom Typ [I_AcsiCommonUnbufferedReportControlBlockClass](#) [▶ 321].

hUser: Optionaler benutzerdefinierter Parameter oder auch Zeiger. Ein beliebiger Wert für diesen Parameter kann in der Methoden für Dienstaktivierung (XyzMethodReq) übergeben werden. In der Ereignisbehandlungsroutine der Dienstrückmeldung wird dann der gleiche Wert gespiegelt.

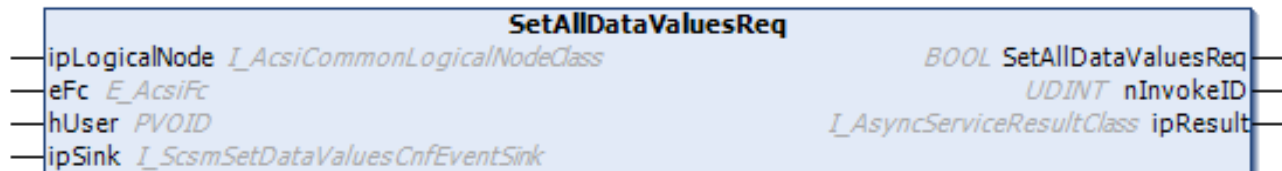
ipSink: Schnittstellenzeiger vom Typ [I_ScsmGetDataValuesCnfEventSink](#) [▶ 379] von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnGetDataValuesCnf (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass](#) [▶ 387]. Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

nInvokeID: Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (XyzMethodReq) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (OnXyzMethodCnf).

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.23 SetAllDataValuesReq



Diese Methode aktiviert den Dienst SetAllDataValues.

```

METHOD SetAllDataValuesReq : BOOL
VAR_INPUT
    ipLogicalNode : I_AcsiCommonLogicalNodeClass;
    eFc           : E_AcsiFc;
    hUser        : PVOID;
    ipSink       : I_ScsmSetDataValuesCnfEventSink;
END_VAR
VAR_OUTPUT
    nInvokeID    : UDINT;
    ipResult     : I_AsyncServiceResultClass;
END_VAR

```

ipLogicalNode: Schnittstellenzeiger vom Typ [I_AcsiCommonLogicalNodeClass](#) [▶ 317].

eFc: Funktionale Gruppe vom Typ [E_AcsiFc](#) [▶ 409].

hUser: Optionaler benutzerdefinierter Parameter oder auch Zeiger. Ein beliebiger Wert für diesen Parameter kann in der Methoden für Dienstaktivierung (XyzMethodReq) übergeben werden. In der Ereignisbehandlungsroutine der Dienstrückmeldung wird dann der gleiche Wert gespiegelt.

ipSink: Schnittstellenzeiger vom Typ [I_ScsmSetDataValuesCnfEventSink \[▶ 385\]](#) von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnSetDataValuesCnf (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass \[▶ 387\]](#). Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

nInvokeID: Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (XyzMethodReq) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (OnXyzMethodCnf).

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.24 SetBrCBValuesReq



Diese Methode aktiviert den Dienst: SetBRCBValues.

```
METHOD SetBRCBValuesReq : BOOL
VAR_INPUT
    ipBrCb      : I_AcsiCommonBufferedReportControlBlockClass;
    stSet       : ST_ScsmBrCBValuesSet;
    hUser       : PVOID;
    ipSink      : I_ScsmSetDataValuesCnfEventSink;
END_VAR
VAR_OUTPUT
    nInvokeID  : UDINT;
    ipResult   : I_AsyncServiceResultClass;
END_VAR
```

ipBrCb: Schnittstellenzeiger vom Typ [I_AcsiCommonBufferedReportControlBlockClass \[▶ 310\]](#).

stSet: Strukturierte Variable des Typs: [ST_ScsmBrCBValuesSet \[▶ 471\]](#). Diese Variable bestimmt welche RCB-Attributwerte geschrieben werden sollen.

hUser: Optionaler benutzerdefinierter Parameter oder auch Zeiger. Ein beliebiger Wert für diesen Parameter kann in der Methoden für Dienstaktivierung (XyzMethodReq) übergeben werden. In der Ereignisbehandlungsroutine der Dienstrückmeldung wird dann der gleiche Wert gespiegelt.

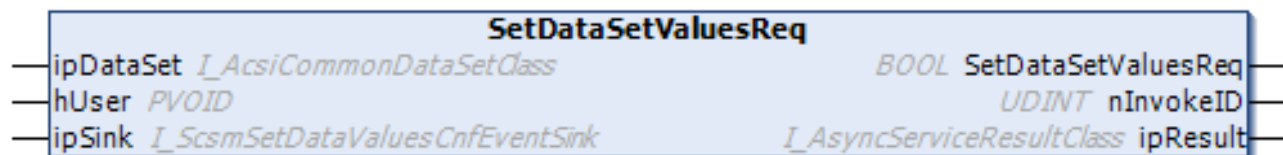
ipSink: Schnittstellenzeiger vom Typ [I_ScsmSetDataValuesCnfEventSink \[▶ 385\]](#) von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnSetDataValuesCnf (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass \[▶ 387\]](#). Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

nInvokeID: Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (XyzMethodReq) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (OnXyzMethodCnf).

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.25 SetDataSetValuesReq



Diese Methode aktiviert den Dienst SetDataSetValues.

```
METHOD SetDataSetValuesReq : BOOL
VAR_INPUT
    ipDataSet : I_AcsiCommonDataSetClass;
    hUser     : PVOID;
    ipSink    : I_ScsmSetDataValuesCnfEventSink;
END_VAR
VAR_OUTPUT
    nInvokeID : UDINT;
    ipResult  : I_AsyncServiceResultClass;
END_VAR
```

ipDataSet: Schnittstellenzeiger vom Typ [I_AcsiCommonDataSetClass](#) [► 313].

hUser: Optionaler benutzerdefinierter Parameter oder auch Zeiger. Ein beliebiger Wert für diesen Parameter kann in der Methoden für Dienstaktivierung (XyzMethodReq) übergeben werden. In der Ereignisbehandlungsroutine der Dienstrückmeldung wird dann der gleiche Wert gespiegelt.

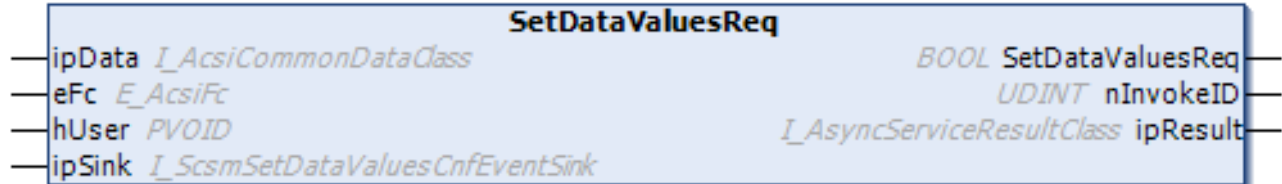
ipSink: Schnittstellenzeiger vom Typ [I_ScsmSetDataValuesCnfEventSink](#) [► 385] von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnSetDataValuesCnf (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass](#) [► 387]. Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

nInvokeID: Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (XyzMethodReq) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (OnXyzMethodCnf).

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.26 SetDataValuesReq



Diese Methode aktiviert den Dienst SetDataValues.

```
METHOD SetDataValuesReq : BOOL
VAR_INPUT
    ipData : I_AcsiCommonDataClass;
    eFc    : E_AcsiFc;
    hUser  : PVOID;
    ipSink : I_ScsmSetDataValuesCnfEventSink;
END_VAR
VAR_OUTPUT
    nInvokeID : UDINT;
    ipResult  : I_AsyncServiceResultClass;
END_VAR
```

ipData: Schnittstellenzeiger vom Typ [I_AcsiCommonDataClass](#) [► 311].

eFc: Funktionale Gruppe vom Typ [E_AcsiFc](#) [► 409].

hUser: Optionaler benutzerdefinierter Parameter oder auch Zeiger. Ein beliebiger Wert für diesen Parameter kann in der Methoden für Dienstaktivierung (XyzMethodReq) übergeben werden. In der Ereignisbehandlungsroutine der Dienstrückmeldung wird dann der gleiche Wert gespiegelt.

ipSink: Schnittstellenzeiger vom Typ [I_ScsmSetDataValuesCnfEventSink](#) [► 385] von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnSetDataValuesCnf (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass](#) [▶ 387]. Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

nInvokeID: Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (XyzMethodReq) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (OnXyzMethodCnf).

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.27 SetGoCBValuesReq



Diese Methode aktiviert den Dienst: SetGOCBValues. Mit dieser Methode kann der Client die Werte eines GOOSE-Steuerblocks (GOOSE Control Block) schreiben.

```

METHOD SetGoCBValuesReq : BOOL
VAR_INPUT
    ipGoCB      : I_AcsiCommonGooseControlBlockClass;
    stSet       : ST_ScsmGoCBValuesSet;
    hUser       : PVOID;
    ipSink      : I_ScsmSetDataValuesCnfEventSink;
END_VAR
VAR_OUTPUT
    nInvokeID   : UDINT;
    ipResult    : I_AsyncServiceResultClass;
END_VAR
    
```

ipGoCB: Schnittstellenzeiger vom Typ [I_AcsiCommonGooseControlBlockClass](#) [▶ 315].

stSet: Strukturierte Variable des Typs: [ST_ScsmGoCBValuesSet](#) [▶ 473]. Diese Variable bestimmt welche GOOSE-Steuerblock Attributwerte geschrieben werden sollen.

hUser: Optionaler benutzerdefinierter Parameter oder auch Zeiger. Ein beliebiger Wert für diesen Parameter kann in der Methoden für Dienstaktivierung (XyzMethodReq) übergeben werden. In der Ereignisbehandlungsroutine der Dienstrückmeldung wird dann der gleiche Wert gespiegelt.

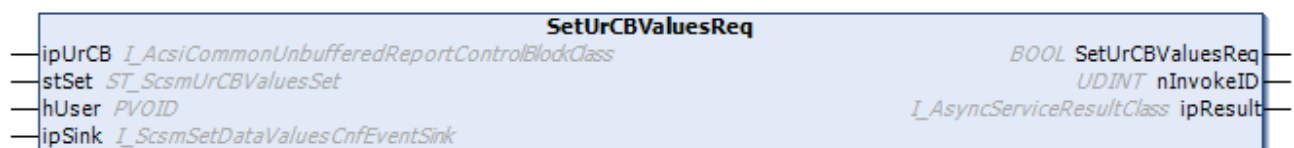
ipSink: Schnittstellenzeiger vom Typ [I_ScsmSetDataValuesCnfEventSink](#) [▶ 385] von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnSetDataValuesCnf (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

nInvokeID: Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (XyzMethodReq) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (OnXyzMethodCnf).

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass](#) [▶ 387]. Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.28 SetUrCBValuesReq



Diese Methode aktiviert den Dienst SetURCBValues.

```

METHOD SetURCBValuesReq : BOOL
VAR_INPUT
  ipUrcb      : I_AcsiCommonUnbufferedReportControlBlockClass;
  stSet       : ST_ScsmUrCBValuesSet;
  hUser       : PVOID;
  ipSink      : I_ScsmSetDataValuesCnfEventSink;
END_VAR
VAR_OUTPUT
  nInvokeID   : UDINT;
  ipResult    : I_AsyncServiceResultClass;
END_VAR

```

ipUrcb: Schnittstellenzeiger vom Typ [I_AcsiCommonUnbufferedReportControlBlockClass](#) [▶ 321].

stSet: Strukturierte Variable des Typs: [ST_ScsmUrCBValuesSet](#) [▶ 475]. Diese Variable bestimmt welche RCB-Attributwerte geschrieben werden sollen.

hUser: Optionaler benutzerdefinierter Parameter oder auch Zeiger. Ein beliebiger Wert für diesen Parameter kann in der Methoden für Dienstaktivierung (XyzMethodReq) übergeben werden. In der Ereignisbehandlungsroutine der Dienstrückmeldung wird dann der gleiche Wert gespiegelt.

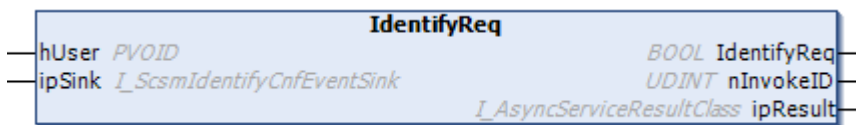
ipSink: Schnittstellenzeiger vom Typ [I_ScsmSetDataValuesCnfEventSink](#) [▶ 385] von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnSetDataValuesCnf (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass](#) [▶ 387]. Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

nInvokeID: Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (XyzMethodReq) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (OnXyzMethodCnf).

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.29 IdentifyReq



Diese Methode aktiviert den Dienst: MMS-Identify.

```

METHOD IdentifyReq : BOOL
VAR_INPUT
  hUser       : PVOID;
  ipSink      : I_ScsmIdentifyCnfEventSink;
END_VAR
VAR_OUTPUT
  nInvokeID   : UDINT;
  ipResult    : I_AsyncServiceResultClass;
END_VAR

```

hUser: Optionaler benutzerdefinierter Parameter oder auch Zeiger. Ein beliebiger Wert für diesen Parameter kann in der Methoden für Dienstaktivierung (XyzMethodReq) übergeben werden. In der Ereignisbehandlungsroutine der Dienstrückmeldung wird dann der gleiche Wert gespiegelt.

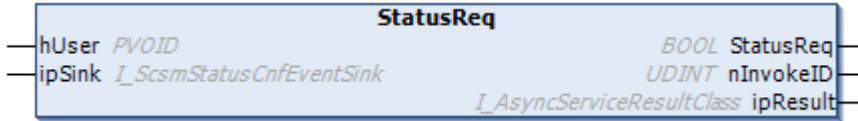
ipSink: Schnittstellenzeiger vom Typ: [I_ScsmIdentifyCnfEventSink](#) [▶ 382] von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnIdentifyCnf (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

nInvokeID: Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (XyzMethodReq) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (OnXyzMethodCnf).

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass](#) [▶ 387]. Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstauführung überwacht und abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.30 StatusReq



Diese Methode aktiviert den Dienst: MMS-Status.

```
METHOD StatusReq : BOOL
VAR_INPUT
    hUser      : PVOID;
    ipSink     : I_ScsmStatusCnfEventSink;
END_VAR
VAR_OUTPUT
    nInvokeID : UDINT;
    ipResult  : I_AsyncServiceResultClass;
END_VAR
```

hUser: Optionaler benutzerdefinierter Parameter oder auch Zeiger. Ein beliebiger Wert für diesen Parameter kann in der Methoden für Dienstaktivierung (XyzMethodReq) übergeben werden. In der Ereignisbehandlungsroutine der Dienstrückmeldung wird dann der gleiche Wert gespiegelt.

ipSink: Schnittstellenzeiger vom Typ: [I_ScsmStatusCnfEventSink \[▶ 386\]](#) von einem Objekt mit einer benutzerdefinierten Implementierung der Ereignisbehandlungsroutine: OnStatusCnf (Dienstrückmeldung). Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

nInvokeID: Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (XyzMethodReq) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (OnXyzMethodCnf).

ipResult: Schnittstellenzeiger vom Typ: [I_AsyncServiceResultClass \[▶ 387\]](#). Über diesen Zeiger kann der Status/Fortgang und das Resultat der Dienstausführung überwacht und abgefragt werden.

Rückgabeparameter: Positive Rückmeldung (TRUE) wenn die Dienstprimitive erfolgreich abgesetzt/ gestartet wurde oder negative Rückmeldung (FALSE) beim Fehler.

6.5.3.6.31 ipIED

Die Instanz der Server-Datenstruktur (oberste Ebene) wird der Instanz des Kommunikationsbausteins über diesen Schnittstellenzeiger bekannt gemacht (mitgeteilt). Über diesen Zeiger kann der Kommunikationsbaustein auf die Datenstruktur lesend oder schreibend zugreifen.

```
PROPERTY ipIED : I_AcsiCommonIntelligentElectronicDeviceClass (SET)
```

ipIED: Schnittstellenzeiger vom Typ [I_AcsiCommonIntelligentElectronicDeviceClass \[▶ 316\]](#) von einem Objekt mit der Implementierung der obersten Ebene des IEC 61850 Datenmodells.

6.5.3.7 I_ScsmControlCnfEventSink

Objekte, die diese Schnittstelle implementieren, besitzen eine benutzerdefinierte Control-Dienst Ereignisbehandlungsroutine für die Schaltersteuerung (Cancel, Operate, Select, SelectWithValue).

Namensraum: [Tc3_iec61850_8_1 \[▶ 97\]](#)

Bibliothek: [Tc3_iec61850_8_1 \(Tc3_iec61850_8_1.compiled-library\)](#)

Vererbungshierarchie

I_ScsmControlCnfEventSink

```
INTERFACE I_ScsmControlCnfEventSink
```

⚡ Methoden zur Ereignisbehandlung

| Name | Beschreibung |
|--------------------------------------|---|
| OnControlCnf [► 376] | Benutzerdefinierte Control-Dienst Ereignisbehandlungsroutine für Schaltersteuerung. |

6.5.3.7.1 OnControlCnf

```
METHOD OnControlCnf
VAR_INPUT
  ipAA      : I_ScsmAssociationClass;
  info      : ST_ScsmConfirmedEventInfo;
  ipCtrl    : I_AcsiCommonControllableDataClass;
  ipData    : I_AcsiCommonDataClass;
  eFc       : E_AcsiFc;
  eError    : E_AcsiServiceError;
END_VAR
```

ipAA: Schnittstellenzeiger vom Typ: [I_ScsmAssociationClass](#) [► 351]. Über diesen Zeiger kann bei mehreren aktiven Verbindungsinstanzen die Dienstausführende Objektinstanz ermittelt werden. Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

info: Strukturierte Variable vom Typ [ST_ScsmConfirmedEventInfo](#) [► 472] mit zusätzlichen Informationen.

ipCtrl: Schnittstellenzeiger vom Typ: [I_AcsiCommonControllableDataClass](#) [► 311] von einem steuerbaren Objekt.

ipData: Schnittstellenzeiger vom Typ [I_AcsiCommonDataClass](#) [► 311].

eFc: Funktionale Gruppe als Aufzählungstyp: [E_AcsiFc](#) [► 409].

eError: Aufzählungstyp: [E_AcsiServiceError](#) [► 428].

6.5.3.8 I_ScsmCommandTerminationIndEventSink

Objekte, die diese Schnittstelle implementieren, besitzen eine benutzerdefinierte Befehlsbeendigung-Dienst Ereignisbehandlungsroutine für die Schaltersteuerung.

Namensraum: [Tc3_iec61850_8_1](#) [► 97]

Bibliothek: [Tc3_iec61850_8_1](#) ([Tc3_iec61850_8_1.compiled-library](#))

Vererbungshierarchie

[I_ScsmCommandTerminationIndEventSink](#)

```
INTERFACE I_ScsmCommandTerminationIndEventSink
```

⚡ Methoden zur Ereignisbehandlung

| Name | Beschreibung |
|---|--|
| OnCommandTerminationInd [► 376] | Benutzerdefinierte Befehlsbeendigung-Dienst Ereignisbehandlungsroutine |

6.5.3.8.1 OnCommandTerminationInd

```
METHOD OnCommandTerminationInd
VAR_INPUT
  ipAA      : I_ScsmAssociationClass;
  info      : ST_ScsmCommandTerminationEventInfo;
  ipCtrl    : I_AcsiCommonControllableDataClass;
  ipData    : I_AcsiCommonDataClass;
  eFc       : E_AcsiFc;
  eError    : E_AcsiServiceError;
END_VAR
```

ipAA: Schnittstellenzeiger vom Typ: [I_ScsmAssociationClass](#) [▶ 351]. Über diesen Zeiger kann bei mehreren aktiven Verbindungsinstanzen die Dienstausführende Objektinstanz ermittelt werden. Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

info: Strukturierte Variable vom Typ [ST_ScsmCommandTerminationEventInfo](#) [▶ 472] mit zusätzlichen Informationen.

ipCtrl: Schnittstellenzeiger vom Typ: [I_AcsiCommonControllableDataClass](#) [▶ 311] von einem steuerbaren Objekt.

ipData: Schnittstellenzeiger vom Typ [I_AcsiCommonDataClass](#) [▶ 311].

eFc: Funktionale Gruppe als Aufzählungstyp: [E_AcsiFc](#) [▶ 409].

eError: Aufzählungstyp: [E_AcsiServiceError](#) [▶ 428].

6.5.3.9 I_ScsmCreateDataSetCnfEventSink

Objekte, die diese Schnittstelle implementieren, besitzen eine benutzerdefinierte CreateDataSet-Dienst Ereignisbehandlungsroutine.

Namensraum: [Tc3_iec61850_8_1](#) [▶ 97]

Bibliothek: [Tc3_iec61850_8_1](#) ([Tc3_iec61850_8_1.compiled-library](#))

Vererbungshierarchie

[I_ScsmCreateDataSetCnfEventSink](#)

INTERFACE [I_ScsmCreateDataSetCnfEventSink](#)

Methoden zur Ereignisbehandlung

| Name | Beschreibung |
|--|---|
| OnCreateDataSetCnf [▶ 377] | Benutzerdefinierte CreateDataSet-Dienst Ereignisbehandlungsroutine. |

6.5.3.9.1 OnCreateDataSetCnf

METHOD [OnCreateDataSetCnf](#)

```
VAR_INPUT
  ipAA      : I_ScsmAssociationClass;
  info     : ST_ScsmConfirmedEventInfo;
  ipDataSet : I_AcsiCommonDataSetClass;
  eError   : E_AcsiServiceError;
END_VAR
```

ipAA: Schnittstellenzeiger vom Typ: [I_ScsmAssociationClass](#) [▶ 351]. Über diesen Zeiger kann bei mehreren aktiven Verbindungsinstanzen die Dienstausführende Objektinstanz ermittelt werden. Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

info: Strukturierte Variable vom Typ [ST_ScsmConfirmedEventInfo](#) [▶ 472] mit zusätzlichen Informationen.

ipDataSet: Schnittstellenzeiger vom Typ: [I_AcsiCommonDataSetClass](#) [▶ 313].

eError: Aufzählungstyp: [E_AcsiServiceError](#) [▶ 428].

6.5.3.10 I_ScsmDeleteDataSetCnfEventSink

Objekte, die diese Schnittstelle implementieren, besitzen eine benutzerdefinierte DeleteDataSet-Dienst Ereignisbehandlungsroutine.

Namensraum: [Tc3_iec61850_8_1](#) [▶ 97]

Bibliothek: [Tc3_iec61850_8_1](#) ([Tc3_iec61850_8_1.compiled-library](#))

Vererbungshierarchie

I_ScsmDeleteDataSetCnfEventSink

INTERFACE I_ScsmDeleteDataSetCnfEventSink

 Methoden zur Ereignisbehandlung

| Name | Beschreibung |
|--|---|
| OnDeleteDataSetCnf [▶ 378] | Benutzerdefinierte DeleteDataSet-Dienst Ereignisbehandlungsroutine. |

6.5.3.10.1 OnDeleteDataSetCnf

METHOD OnDeleteDataSetCnf

```
VAR_INPUT
  ipAA      : I_ScsmAssociationClass;
  info      : ST_ScsmConfirmedEventInfo;
  ipDataSet : I_AcsiCommonDataSetClass;
  eError    : E_AcsiServiceError;
END_VAR
```

ipAA: Schnittstellenzeiger vom Typ: [I_ScsmAssociationClass](#) [[▶ 351](#)]. Über diesen Zeiger kann bei mehreren aktiven Verbindungsinstanzen die Dienstausführende Objektinstanz ermittelt werden. Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

info: Strukturierte Variable vom Typ [ST_ScsmConfirmedEventInfo](#) [[▶ 472](#)] mit zusätzlichen Informationen.

ipDataSet: Schnittstellenzeiger vom Typ: [I_AcsiCommonDataSetClass](#) [[▶ 313](#)].

eError: Aufzählungstyp: [E_AcsiServiceError](#) [[▶ 428](#)].

6.5.3.11 I_ScsmGetDataSetDirectoryCnfEventSink

Objekte, die diese Schnittstelle implementieren, besitzen eine benutzerdefinierte GetDataSetDirectory-Dienst Ereignisbehandlungsroutine.

Namensraum: [Tc3_iec61850_8_1](#) [[▶ 97](#)]

Bibliothek: [Tc3_iec61850_8_1](#) ([Tc3_iec61850_8_1.compiled-library](#))

Vererbungshierarchie

I_ScsmGetDataSetDirectoryCnfEventSink

INTERFACE I_ScsmGetDataSetDirectoryCnfEventSink

 Methoden zur Ereignisbehandlung

| Name | Beschreibung |
|--|--|
| OnGetDataSetDirectoryCnf [▶ 378] | Benutzerdefinierte GetDataSetDirectory-Dienst Ereignisbehandlungsroutine |

6.5.3.11.1 OnGetDataSetDirectoryCnf

METHOD OnGetDataSetDirectoryCnf

```
VAR_INPUT
  ipAA      : I_ScsmAssociationClass;
  info      : ST_ScsmConfirmedEventInfo;
  ipDataSet : I_AcsiCommonDataSetClass;
  sObjectReference : T_AcsiObjectReference;
  eFc       : E_AcsiFc;
  eError    : E_AcsiServiceError;
END_VAR
```

ipAA: Schnittstellenzeiger vom Typ: [I_ScsmAssociationClass](#) [[▶ 351](#)]. Über diesen Zeiger kann bei mehreren aktiven Verbindungsinstanzen die Dienstausführende Objektinstanz ermittelt werden. Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

info: Strukturierte Variable vom Typ [ST_ScsmConfirmedEventInfo](#) [▶ 472] mit zusätzlichen Informationen.

ipDataSet: Schnittstellenzeiger vom Typ: [I_AcsiCommonDataSetClass](#) [▶ 313].

sObjectReference:

eFc: Funktionale Gruppe als Aufzählungstyp: [E_AcsiFc](#) [▶ 409].

eError: Aufzählungstyp: [E_AcsiServiceError](#) [▶ 428].

6.5.3.12 I_ScsmGetDataValuesCnfEventSink

Objekte, die diese Schnittstelle implementieren, besitzen eine benutzerdefinierte GetDataValues-Dienst Ereignisbehandlungsroutine.

Namensraum: [Tc3_iec61850_8_1](#) [▶ 97]

Bibliothek: [Tc3_iec61850_8_1](#) ([Tc3_iec61850_8_1.compiled-library](#))

Vererbungshierarchie

[I_ScsmGetDataValuesCnfEventSink](#)

INTERFACE [I_ScsmGetDataValuesCnfEventSink](#)

Methoden zur Ereignisbehandlung

| Name | Beschreibung |
|---|--|
| OnGetDataValuesCnf
[▶ 379] | Benutzerdefinierte GetDataValues-Dienst Ereignisbehandlungsroutine |

6.5.3.12.1 OnGetDataValuesCnf

```
METHOD OnGetDataValuesCnf
VAR_INPUT
    ipAA    : I_ScsmAssociationClass;
    info    : ST_ScsmConfirmedEventInfo;
    ipData  : I_AcsiCommonDataClass;
    eFc     : E_AcsiFc;
    eError  : E_AcsiServiceError;
END_VAR
```

ipAA: Schnittstellenzeiger vom Typ: [I_ScsmAssociationClass](#) [▶ 351]. Über diesen Zeiger kann bei mehreren aktiven Verbindungsinstanzen die Dienstausführende Objektinstanz ermittelt werden. Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

info: Strukturierte Variable vom Typ [ST_ScsmConfirmedEventInfo](#) [▶ 472] mit zusätzlichen Informationen.

ipData: Schnittstellenzeiger vom Typ: [I_AcsiCommonDataClass](#) [▶ 311].

eFc: Funktionale Gruppe als Aufzählungstyp: [E_AcsiFc](#) [▶ 409].

eError: Aufzählungstyp: [E_AcsiServiceError](#) [▶ 428].

6.5.3.13 I_ScsmGetLogicalDeviceDirectoryCnfEventSink

Objekte, die diese Schnittstelle implementieren, besitzen eine benutzerdefinierte GetLogicalDeviceDirectory-Dienst Ereignisbehandlungsroutine.

Namensraum: [Tc3_iec61850_8_1](#) [▶ 97]

Bibliothek: [Tc3_iec61850_8_1](#) ([Tc3_iec61850_8_1.compiled-library](#))

Vererbungshierarchie

[I_ScsmGetLogicalDeviceDirectoryCnfEventSink](#)


```
INTERFACE I_ScsmGetLogicalDeviceDirectoryCnfEventSink
```

Methoden zur Ereignisbehandlung

| Name | Beschreibung |
|--|--|
| OnGetLogicalDeviceDirectoryCnf [▶ 380] | Benutzerdefinierte GetLogicalDeviceDirectory-Dienst Ereignisbehandlungsroutine |

6.5.3.13.1 OnGetLogicalDeviceDirectoryCnf

```
METHOD OnGetLogicalDeviceDirectoryCnf
VAR_INPUT
    ipAA          : I_ScsmAssociationClass;
    info          : ST_ScsmConfirmedEventInfo;
    ipLogicalDevice : I_AcsiCommonLogicalDeviceClass;
    sObjectReference : T_AcsiObjectReference;
    eError        : E_AcsiServiceError;
END_VAR
```

ipAA: Schnittstellenzeiger vom Typ: [I_ScsmAssociationClass](#) [[▶ 351](#)]. Über diesen Zeiger kann bei mehreren aktiven Verbindungsinstanzen die Dienstausführende Objektinstanz ermittelt werden. Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

info: Strukturierte Variable vom Typ [ST_ScsmConfirmedEventInfo](#) [[▶ 472](#)] mit zusätzlichen Informationen.

ipLogicalDevice: Schnittstellenzeiger vom Typ [I_AcsiCommonLogicalDeviceClass](#) [[▶ 317](#)].

sObjectReference:

eError: Aufzählungstyp: [E_AcsiServiceError](#) [[▶ 428](#)].

6.5.3.14 I_ScsmGetLogicalNodeDirectoryCnfEventSink

Objekte, die diese Schnittstelle implementieren besitzen eine benutzerdefinierte GetLogicalNodeDirectory-Dienst Ereignisbehandlungsroutine.

Namensraum: [Tc3_iec61850_8_1](#) [[▶ 97](#)]

Bibliothek: [Tc3_iec61850_8_1](#) ([Tc3_iec61850_8_1.compiled-library](#))

Vererbungshierarchie

[I_ScsmGetLogicalNodeDirectoryCnfEventSink](#)

```
INTERFACE I_ScsmGetLogicalNodeDirectoryCnfEventSink
```

Methoden zur Ereignisbehandlung

| Name | Beschreibung |
|--|--|
| OnGetLogicalNodeDirectoryCnf [▶ 380] | Benutzerdefinierte GetLogicalNodeDirectory-Dienst Ereignisbehandlungsroutine |

6.5.3.14.1 OnGetLogicalNodeDirectoryCnf

```
METHOD OnGetLogicalNodeDirectoryCnf
VAR_INPUT
    ipAA          : I_ScsmAssociationClass;
    info          : ST_ScsmConfirmedEventInfo;
    ipLogicalNode  : I_AcsiCommonLogicalNodeClass;
    eClass        : E_AcsiLogicalNodeClass;
    sObjectReference : T_AcsiObjectReference;
    eError        : E_AcsiServiceError;
END_VAR
```


ipAA: Schnittstellenzeiger vom Typ: [I_ScsmAssociationClass](#) [▶ 351]. Über diesen Zeiger kann bei mehreren aktiven Verbindungsinstanzen die Dienstaufführende Objektinstanz ermittelt werden. Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

info: Strukturierte Variable vom Typ [ST_ScsmConfirmedEventInfo](#) [▶ 472] mit zusätzlichen Informationen.

ipLogicalNode: Schnittstellenzeiger vom Typ [I_AcsiCommonLogicalNodeClass](#) [▶ 317].

eClass: Aufzählungsvariable vom Typ [E_AcsiLogicalNodeClass](#) [▶ 414].

sObjectReference:

eError: Aufzählungstyp: [E_AcsiServiceError](#) [▶ 428].

6.5.3.15 I_ScsmGetServerDirectoryCnfEventSink

Objekte, die diese Schnittstelle implementieren, besitzen eine benutzerdefinierte GetServerDirectory-Dienst Ereignisbehandlungsroutine.

Namensraum: [Tc3_iec61850_8_1](#) [▶ 97]

Bibliothek: [Tc3_iec61850_8_1](#) ([Tc3_iec61850_8_1.compiled-library](#))

Vererbungshierarchie

[I_ScsmGetServerDirectoryCnfEventSink](#)

INTERFACE [I_ScsmGetServerDirectoryCnfEventSink](#)

Methoden zur Ereignisbehandlung

| Name | Beschreibung |
|--|---|
| OnGetServerDirectoryCnf
[▶ 381] | Benutzerdefinierte GetServerDirectory-Dienst Ereignisbehandlungsroutine |

6.5.3.15.1 OnGetServerDirectoryCnf

```
METHOD OnGetServerDirectoryCnf
VAR_INPUT
    ipAA          : I_ScsmAssociationClass;
    info          : ST_ScsmConfirmedEventInfo;
    ipServer      : I_AcsiCommonIntelligentElectronicDeviceClass;
    eClass        : E_AcsiServerDirectoryClass;
    sObjectReference : T_AcsiObjectReference;
    eError        : E_AcsiServiceError;
END_VAR
```

ipAA: Schnittstellenzeiger vom Typ: [I_ScsmAssociationClass](#) [▶ 351]. Über diesen Zeiger kann bei mehreren aktiven Verbindungsinstanzen die Dienstaufführende Objektinstanz ermittelt werden. Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

info: Strukturierte Variable vom Typ [ST_ScsmConfirmedEventInfo](#) [▶ 472] mit zusätzlichen Informationen.

ipServer: Schnittstellenzeiger vom Typ [I_AcsiCommonIntelligentElectronicDeviceClass](#) [▶ 316] vom Server-Objekt, der abgefragt wird.

eClass: Aufzählungsvariable vom Typ [E_AcsiServerDirectoryClass](#) [▶ 428]. Dieser Parameter gibt an, welche Information abgefragt wird: Logische Knoten oder Dateien.

sObjectReference:

eError: Aufzählungstyp: [E_AcsiServiceError](#) [▶ 428].

6.5.3.16 I_ScsmIdentifyCnfEventSink

Objekte, die diese Schnittstelle implementieren, besitzen eine benutzerdefinierte Identify-Dienst Ereignisbehandlungsroutine.

Namensraum: [Tc3_iec61850_8_1](#) [[▶ 97](#)]

Bibliothek: [Tc3_iec61850_8_1](#) ([Tc3_iec61850_8_1.compiled-library](#))

Vererbungshierarchie

I_ScsmIdentifyCnfEventSink

```
INTERFACE I_ScsmIdentifyCnfEventSink
```

Methoden zur Ereignisbehandlung

| Name | Beschreibung |
|---|--|
| OnIdentifyCnf [▶ 382] | Benutzerdefinierte Identify-Dienst Ereignisbehandlungsroutine. |

6.5.3.16.1 OnIdentifyCnf

```
METHOD OnIdentifyCnf
VAR_INPUT
    ipAA      : I_ScsmAssociationClass;
    info      : ST_ScsmConfirmedEventInfo;
    sVendorName : T_MmsString;
    sModelName : T_MmsString;
    sRevision  : T_MmsString;
    eError     : E_AcsiServiceError;
END_VAR
```

ipAA: Schnittstellenzeiger vom Typ: [I_ScsmAssociationClass](#) [[▶ 351](#)]. Über diesen Zeiger kann bei mehreren aktiven Verbindungsinstanzen die Dienstausführende Objektinstanz ermittelt werden. Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

info: Strukturierte Variable vom Typ [ST_ScsmConfirmedEventInfo](#) [[▶ 472](#)] mit zusätzlichen Informationen.

sVendorName: Herstellerbezeichnung vom Typ: [T_MmsString](#) [[▶ 478](#)].

sModelName: Modellbezeichnung vom Typ: [T_MmsString](#) [[▶ 478](#)].

sRevision: Revisionsstring vom Typ: [T_MmsString](#) [[▶ 478](#)].

eError: Aufzählungstyp: [E_AcsiServiceError](#) [[▶ 428](#)].

6.5.3.17 I_ScsmLastAppLErrorIndEventSink

Objekte, die diese Schnittstelle implementieren, besitzen eine benutzerdefinierte LastAppLError-Dienst Ereignisbehandlungsroutine für den Zugriff auf zusätzliche Report-Diagnosedaten.

Namensraum: [Tc3_iec61850_8_1](#) [[▶ 97](#)]

Bibliothek: [Tc3_iec61850_8_1](#) ([Tc3_iec61850_8_1.compiled-library](#))

Vererbungshierarchie

I_ScsmLastAppLErrorIndEventSink

```
INTERFACE I_ScsmLastAppLErrorIndEventSink
```

Methoden zur Ereignisbehandlung

| Name | Beschreibung |
|--|--|
| OnLastAppLErrorInd [▶ 383] | Benutzerdefinierte LastAppLError-Dienst Ereignisbehandlungsroutine |

6.5.3.17.1 OnLastApplErrorInd

```
METHOD OnLastApplErrorInd
VAR_INPUT
    ipAA : I_ScsmAssociationClass;
    info : ST_ScsmAdditionalCauseDiagnosticEventInfo;
    ipCtrl : I_AcsiCommonControllableDataClass;
    eError : E_AcsiServiceError;
END_VAR
```

ipAA: Schnittstellenzeiger vom Typ: [I_ScsmAssociationClass \[▶ 351\]](#). Über diesen Zeiger kann bei mehreren aktiven Verbindungsinstanzen die Dienstaufführende Objektinstanz ermittelt werden. Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

info: Strukturierte Variable vom Typ [ST_ScsmAdditionalCauseDiagnosticEventInfo \[▶ 471\]](#) mit zusätzlichen Informationen zum LastApplError.

ipCtrl: Schnittstellenzeiger vom Typ: [I_AcsiCommonControllableDataClass \[▶ 311\]](#) von einem steuerbaren Objekt.

eError: Aufzählungstyp: [E_AcsiServiceError \[▶ 428\]](#).

6.5.3.18 I_ScsmReleaseCnfEventSink

Objekte, die diese Schnittstelle implementieren, besitzen eine benutzerdefinierte Release-Dienst Ereignisbehandlungsroutine für den Verbindungsabbau.

Namensraum: [Tc3_iec61850_8_1 \[▶ 97\]](#)

Bibliothek: [Tc3_iec61850_8_1 \(Tc3_iec61850_8_1.compiled-library\)](#)

Vererbungshierarchie

I_ScsmReleaseCnfEventSink

```
INTERFACE I_ScsmReleaseCnfEventSink
```

Methoden zur Ereignisbehandlung

| Name | Beschreibung |
|--------------------------------------|--|
| OnReleaseCnf [▶ 383] | Benutzerdefinierte Release-Dienst Ereignisbehandlungsroutine |

6.5.3.18.1 OnReleaseCnf

```
METHOD OnReleaseCnf
VAR_INPUT
    ipAA : I_ScsmAssociationClass;
    eError : E_AcsiServiceError;
END_VAR
```

ipAA: Schnittstellenzeiger vom Typ: [I_ScsmAssociationClass \[▶ 351\]](#). Über diesen Zeiger kann bei mehreren aktiven Verbindungsinstanzen die Dienstaufführende Objektinstanz ermittelt werden. Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

eError: Aufzählungstyp: [E_AcsiServiceError \[▶ 428\]](#).

6.5.3.19 I_ScsmReleaseIndEventSink

Objekte, die diese Schnittstelle implementieren, besitzen eine benutzerdefinierte Release-Dienst Ereignisbehandlungsroutine für den Verbindungsabbau.

Namensraum: [Tc3_iec61850_8_1 \[▶ 97\]](#)

Bibliothek: [Tc3_iec61850_8_1 \(Tc3_iec61850_8_1.compiled-library\)](#)

Vererbungshierarchie

I_ScsmReleaseIndEventSink

INTERFACE I_ScsmReleaseIndEventSink

⚡ Methoden zur Ereignisbehandlung

| Name | Beschreibung |
|--|--|
| OnReleaseInd [▶ 384] | Benutzerdefinierte Release-Dienst Ereignisbehandlungsroutine |

6.5.3.19.1 OnReleaseInd

```
METHOD OnReleaseInd : E_EventCompletion
VAR_INPUT
    ipAA : I_ScsmAssociationClass;
END_VAR
```

ipAA: Schnittstellenzeiger vom Typ: [I_ScsmAssociationClass](#) [[▶ 351](#)]. Über diesen Zeiger kann bei mehreren aktiven Verbindungsinstanzen die Dienstaufführende Objektinstanz ermittelt werden. Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

6.5.3.20 I_ScsmReportIndEventSink

Objekte, die diese Schnittstelle implementieren, besitzen eine benutzerdefinierte Report-Dienst Ereignisbehandlungsroutine.

Namensraum: [Tc3_iec61850_8_1](#) [[▶ 97](#)]

Bibliothek: [Tc3_iec61850_8_1](#) ([Tc3_iec61850_8_1.compiled-library](#))

Vererbungshierarchie

I_ScsmReportIndEventSink

INTERFACE I_ScsmReportIndEventSink

⚡ Methoden zur Ereignisbehandlung

| Name | Beschreibung |
|--|--|
| OnBufferedReportInd
▶ 384 | Benutzerdefinierte Buffered-Report-Dienst Ereignisbehandlungsroutine |
| OnUnbufferedReportInd
▶ 385 | Benutzerdefinierte Unbuffered-Report-Dienst Ereignisbehandlungsroutine |

6.5.3.20.1 OnBufferedReportInd

```
METHOD OnBufferedReportInd
VAR_INPUT
    ipAA : I_ScsmAssociationClass;
    info : ST_ScsmReportEventInfo;
    ipBrCb : I_AcsiCommonBufferedReportControlBlockClass;
    ipDataSet : I_AcsiCommonDataSetClass;
    ipData : I_AcsiCommonDataClass;
    eFc : E_AcsiFc;
    eError : E_AcsiServiceError;
END_VAR
```

ipAA: Schnittstellenzeiger vom Typ: [I_ScsmAssociationClass](#) [[▶ 351](#)]. Über diesen Zeiger kann bei mehreren aktiven Verbindungsinstanzen die Dienstaufführende Objektinstanz ermittelt werden. Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

info: Strukturierte Variable vom Typ [ST_ScsmReportEventInfo](#) [[▶ 474](#)] mit zusätzlichen Informationen zum Report.

ipBrCb: Schnittstellenzeiger vom Typ [I_AcsiCommonBufferedReportControlBlockClass](#) [[▶ 310](#)].

ipDataSet: Schnittstellenzeiger vom Typ: [I_AcsiCommonDataSetClass](#) [[▶ 313](#)].

ipData: Schnittstellenzeiger vom Typ: [I_AcsiCommonDataClass](#) [▶ 311].

eFc: Funktionale Gruppe als Aufzählungstyp: [E_AcsiFc](#) [▶ 409].

eError: Aufzählungstyp: [E_AcsiServiceError](#) [▶ 428].

6.5.3.20.2 OnUnbufferedReportInd

```
METHOD OnUnbufferedReportInd
VAR_INPUT
    ipAA      : I_ScsmAssociationClass;
    info      : ST_ScsmReportEventInfo;
    ipUrcb    : I_AcsiCommonUnbufferedReportControlBlockClass;
    ipDataSet : I_AcsiCommonDataSetClass;
    ipData    : I_AcsiCommonDataClass;
    eFc       : E_AcsiFc;
    eError    : E_AcsiServiceError;
END_VAR
```

ipAA: Schnittstellenzeiger vom Typ: [I_ScsmAssociationClass](#) [▶ 351]. Über diesen Zeiger kann bei mehreren aktiven Verbindungsinstanzen die Dienstausführende Objektinstanz ermittelt werden. Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

info: Strukturierte Variable vom Typ [ST_ScsmReportEventInfo](#) [▶ 474] mit zusätzlichen Informationen zum Report.

ipUrcb: Schnittstellenzeiger vom Typ [I_AcsiCommonUnbufferedReportControlBlockClass](#) [▶ 321].

ipDataSet: Schnittstellenzeiger vom Typ: [I_AcsiCommonDataSetClass](#) [▶ 313].

ipData: Schnittstellenzeiger vom Typ: [I_AcsiCommonDataClass](#) [▶ 311].

eFc: Funktionale Gruppe als Aufzählungstyp: [E_AcsiFc](#) [▶ 409].

eError: Aufzählungstyp: [E_AcsiServiceError](#) [▶ 428].

6.5.3.21 I_ScsmSetDataValuesCnfEventSink

Objekte, die diese Schnittstelle implementieren, besitzen eine benutzerdefinierte SetDataValues-Dienst Ereignisbehandlungsroutine.

Namensraum: [Tc3_iec61850_8_1](#) [▶ 97]

Bibliothek: [Tc3_iec61850_8_1](#) (Tc3_iec61850_8_1.compiled-library)

Vererbungshierarchie

I_ScsmSetDataValuesCnfEventSink

```
INTERFACE I_ScsmSetDataValuesCnfEventSink
```

Methoden zur Ereignisbehandlung

| Name | Beschreibung |
|---|--|
| OnSetDataValuesCnf
[▶ 385] | Benutzerdefinierte SetDataValues-Dienst Ereignisbehandlungsroutine |

6.5.3.21.1 OnSetDataValuesCnf

```
METHOD OnSetDataValuesCnf
VAR_INPUT
    ipAA      : I_ScsmAssociationClass;
    info      : ST_ScsmConfirmedEventInfo;
    ipData    : I_AcsiCommonDataClass;
    eFc       : E_AcsiFc;
    eError    : E_AcsiServiceError;
END_VAR
```

ipAA: Schnittstellenzeiger vom Typ: [I_ScsmAssociationClass](#) [▶ 351]. Über diesen Zeiger kann bei mehreren aktiven Verbindungsinstanzen die Dienstaufführende Objektinstanz ermittelt werden. Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

info: Strukturierte Variable vom Typ [ST_ScsmConfirmedEventInfo](#) [▶ 472] mit zusätzlichen Informationen.

ipData: Schnittstellenzeiger vom Typ: [I_AcsiCommonDataClass](#) [▶ 311].

eFc: Funktionale Gruppe als Aufzählungstyp: [E_AcsiFc](#) [▶ 409].

eError: Aufzählungstyp: [E_AcsiServiceError](#) [▶ 428].

6.5.3.22 I_ScsmStatusCnfEventSink

Objekte, die diese Schnittstelle implementieren, besitzen eine benutzerdefinierte Status-Dienst Ereignisbehandlungsroutine.

Namensraum: [Tc3_iec61850_8_1](#) [▶ 97]

Bibliothek: [Tc3_iec61850_8_1](#) ([Tc3_iec61850_8_1.compiled-library](#))

Vererbungshierarchie

I_ScsmStatusCnfEventSink

INTERFACE I_ScsmStatusCnfEventSink

Methoden zur Ereignisbehandlung

| Name | Beschreibung |
|-------------------------------------|--|
| OnStatusCnf [▶ 386] | Benutzerdefinierte Status-Dienst Ereignisbehandlungsroutine. |

6.5.3.22.1 OnStatusCnf

```
METHOD OnStatusCnf
VAR_INPUT
    ipAA          : I_ScsmAssociationClass;
    info          : ST_ScsmConfirmedEventInfo;
    eVmdLogicalStatus : E_MmsVmdLogicalStatus;
    eVmdPhysicalStatus : E_MmsVmdPhysicalStatus;
    ipLocalDetail  : I_MmsBitString;
    eError        : E_AcsiServiceError;
END_VAR
```

ipAA: Schnittstellenzeiger vom Typ: [I_ScsmAssociationClass](#) [▶ 351]. Über diesen Zeiger kann bei mehreren aktiven Verbindungsinstanzen die Dienstaufführende Objektinstanz ermittelt werden. Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

info: Strukturierte Variable vom Typ [ST_ScsmConfirmedEventInfo](#) [▶ 472] mit zusätzlichen Informationen.

eVmdLogicalStatus: Aufzählungstyp: [E_MmsVmdLogicalStatus](#) [▶ 447].

eVmdPhysicalStatus: Aufzählungstyp: [E_MmsVmdPhysicalStatus](#) [▶ 447].

ipLocalDetail: Schnittstellenzeiger vom Typ: [I_MmsBitString](#).

eError: Aufzählungstyp: [E_AcsiServiceError](#) [▶ 428].

6.5.3.23 I_ScsmSystemClockEventSink

Objekte, die diese Schnittstelle implementieren, besitzen eine benutzerdefinierte SystemClock-Dienst Ereignisbehandlungsroutine für den Zugriff auf eine externe Zeitquelle.

Namensraum: [Tc3_iec61850_8_1](#) [▶ 97]

Bibliothek: [Tc3_iec61850_8_1](#) ([Tc3_iec61850_8_1.compiled-library](#))

Vererbungshierarchie

I_ScsmSystemClockEventSink

INTERFACE I_ScsmSystemClockEventSink

Methoden zur Ereignisbehandlung

| Name | Beschreibung |
|---|--|
| <u>OnGetSystemTime</u>
▶ 387 | Benutzerdefinierte SystemClock-Dienst Ereignisbehandlungsroutine |

6.5.3.23.1 OnGetSystemTime

```
METHOD OnGetSystemTime : BOOL
VAR_INPUT
    ipAA      : I_ScsmAssociationClass;
END_VAR
VAR_OUTPUT
    utcTime   : T_UtcTime;
END_VAR
```

ipAA: Schnittstellenzeiger vom Typ: [I_ScsmAssociationClass](#) [▶ 351](#). Über diesen Zeiger kann bei mehreren aktiven Verbindungsinstanzen die Dienstausführende Objektinstanz ermittelt werden. Dieser Parameter ist optional und für zukünftige Verwendung reserviert. Der Zeigerwert = 0 definiert den Parameter als optional.

utcTime: Aktuelle Utc-Time-Zeitinformation vom Typ [T_UtcTime](#) [▶ 479](#).

6.5.4 I_AsyncServiceResultClass

Generische Schnittstelle zur Statusabfrage und Verfolgung von asynchronen Vorgängen. Dazu gehören Dienste, die für die Ausführung mehrere SPS-Zyklen benötigen. Bei Objekten, die diese Schnittstelle implementieren, kann der Status/Fortgang und das Resultat der Dienstausführung überwacht und abgefragt werden.

Namensraum: [Tc3_Collections](#) [▶ 96](#)

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Vererbungshierarchie

I_AsyncServiceResultClass

INTERFACE I_AsyncServiceResultClass

Methoden

| Name | Beschreibung |
|--------------------------------------|--|
| CopyFrom [▶ 388] | Kopiert Objektdaten von einer anderen Objektinstanz |
| Dump [▶ 388] | Konvertiert eigene Objektdaten in einen formatierten String |
| Equal [▶ 388] | Wertevergleich der eigenen Objektdaten mit externen Datenwerten |
| EqualEx [▶ 389] | Wertevergleich der eigenen Objektdaten mit externen Datenwerten |
| EqualTo [▶ 389] | Wertevergleich der eigenen Objektdaten mit den Daten einer anderen Objektinstanz |
| IsAborted [▶ 389] | Prüft, ob die Dienstaufführung abgebrochen wurde |
| IsBusy [▶ 389] | Prüft, ob der Dienst mit der Ausführung noch beschäftigt ist |
| IsCompleted [▶ 390] | Prüft, ob die Dienstaufführung erfolgreich abgeschlossen wurde |
| IsFailed [▶ 390] | Prüft, ob die Dienstaufführung mit einem Fehler abgeschlossen wurde |
| SetAborted [▶ 390] | Setzt den Abbruchstatus |
| SetAbortedEx [▶ 390] | Setzt den Abbruchstatus |
| SetBusy [▶ 391] | Setzt den Status auf Beschäftigt |
| SetCompleted [▶ 391] | Setzt den Status auf Erledigt (ohne Fehler) |
| SetFailed [▶ 391] | Setzt den Fehlerstatus |
| SetFailedEx [▶ 392] | Setzt den Fehlerstatus |

Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|--------------------------------|---|---------|-------------------|
| eState [▶ 392] | E AsyncServiceState [▶ 444] | Get | Statusinformation |

6.5.4.1 CopyFrom

Kopiert Objektdaten von einer anderen Objektinstanz.

```
METHOD CopyFrom : BOOL
VAR_INPUT
    ipSource : I_AsyncServiceResultClass;
END_VAR
```

ipSource: Schnittstellenzeiger vom Typ [I_AsyncServiceResultClass \[▶ 387\]](#) von der Objektinstanz der Datenquelle.

Rückgabeparameter: TRUE => Erfolg, FALSE => Fehler (z. B. Schnittstellenzeiger ist Null).

6.5.4.2 Dump

Konvertiert eigene Objektdaten in einen formatierten String (z. B. für Logausgaben).

```
METHOD Dump : T_MaxString
```

Rückgabeparameter: String mit dem folgenden Format: „{State: {Severity: [Verbose][Info][Warning][Error][Critical], Id: n[nnnnn], Class: „NameOrGuid“}}“.

6.5.4.3 Equal

Wertevergleich der eigenen Objektdaten (Status und Fehlerinformation) mit externen Datenwerten.


```
METHOD Equal : BOOL
VAR_INPUT
    eOther : E_AsyncServiceState;
    ipOther : I_ServiceErrorClass;
END_VAR
```

eOther: Aufzählungstyp [E_AsyncServiceState](#) [[▶ 444](#)] mit dem zu vergleichenden Status.

ipOther: Schnittstellenzeiger vom Typ [I_ServiceErrorClass](#) [[▶ 392](#)] von der Objektinstanz mit der zu vergleichenden Fehlerinformation.

Rückgabeparameter: TRUE => Daten sind gleich, FALSE => Daten sind unterschiedlich.

6.5.4.4 EqualEx

Wertevergleich der eigenen Objektdaten (Status und Fehlerinformation) mit externen Datenwerten.

```
METHOD EqualEx : BOOL
VAR_INPUT
    eOther : E_AsyncServiceState;
    stOther : ST_ServiceErrorClass;
END_VAR
```

eOther: Aufzählungstyp [E_AsyncServiceState](#) [[▶ 444](#)] mit dem zu vergleichenden Status.

stOther: Strukturierte Variable vom Typ [ST_ServiceErrorClass](#) [[▶ 475](#)] mit der zu vergleichenden Fehlerinformation.

Rückgabeparameter: TRUE => Daten sind gleich, FALSE => Daten sind unterschiedlich.

6.5.4.5 EqualTo

Wertevergleich der eigenen Objektdaten (Status und Fehlerinformation) mit den Daten einer anderen Objektinstanz.

```
METHOD EqualTo : BOOL
VAR_INPUT
    ipOther : I_AsyncServiceResultClass;
END_VAR
```

ipOther: Schnittstellenzeiger vom Typ [I_AsyncServiceResultClass](#) [[▶ 387](#)] von der Objektinstanz, deren Daten verglichen werden sollen.

Rückgabeparameter: TRUE => Daten sind gleich, FALSE => Daten sind unterschiedlich.

6.5.4.6 IsAborted

Prüft ob die Dienstauführung abgebrochen wurde.

```
METHOD IsAborted : BOOL
```

Rückgabeparameter: TRUE => Abgebrochen, FALSE => Nicht abgebrochen (Busy oder Completed oder Failed).

6.5.4.7 IsBusy

Prüft, ob der Dienst mit der Ausführung noch beschäftigt ist.

```
METHOD IsBusy : BOOL
```

Rückgabeparameter: TRUE => Beschäftigt, FALSE => Nicht beschäftigt (Completed oder Aborted oder Failed).

Beispiel (Auszug)

```
METHOD MyServiceReq : BOOL
VAR_INPUT
    tTimeout : TIME;
END_VAR
```

```

VAR_OUTPUT
    ipResult : I_AsyncServiceResultClass := _fbResult;
END_VAR

IF ipResult.IsBusy() THEN(* service already running => abort *)
    _state := 0;
    ipResult.SetAbortedEx( stReason := ERROR_INVALID_STATE );
    RETURN;
END_IF
IF tTimeout = T#0S THEN(* invalid parameter => failure *)
    ipResult.SetFailedEx( stError := ERROR_INVALID_VALUE );
    RETURN;
END_IF
ipResult.SetBusy();
_fbTimer( IN := FALSE, PT := tTimeout );
_state := 1;
THIS^();
MyServiceReq := TRUE;

```

6.5.4.8 IsCompleted

Prüft, ob die Dienstauführung erfolgreich abgeschlossen wurde.

```
METHOD IsCompleted : BOOL
```

Rückgabeparameter: TRUE => Abgeschlossen ohne Fehler. FALSE => Nicht abgeschlossen (Busy oder Aborted oder Failed).

6.5.4.9 IsFailed

Prüft, ob die Dienstauführung mit einem Fehler abgeschlossen wurde.

```
METHOD IsFailed : BOOL
```

Rückgabeparameter: TRUE => Abgeschlossen mit einem Fehler, FALSE => Kein Abschluss mit einem Fehler (Busy oder Completed oder Aborted).

6.5.4.10 SetAborted

Setzt den Abbruchstatus und speichert die Daten zur Abbruchursache in der eigenen Objektinstanz.

```

METHOD SetAborted : I_AsyncServiceResultClass
VAR_INPUT
    ipReason : I_ServiceErrorClass;
END_VAR

```

ipReason: Schnittstellenzeiger vom Typ [I_ServiceErrorClass](#) [▶ 392] von einer Objektinstanz mit den Daten zur Abbruchursache.

Rückgabeparameter: Eigener Schnittstellenzeiger vom Typ [I_AsyncServiceResultClass](#) [▶ 387].

6.5.4.11 SetAbortedEx

Setzt den Abbruchstatus und speichert die Daten zur Abbruchursache in der eigenen Objektinstanz.

```

METHOD SetAbortedEx : I_AsyncServiceResultClass
VAR_INPUT
    stReason : ST_ServiceErrorClass;
END_VAR

```

stReason: Strukturierte Variable vom Typ [ST_ServiceErrorClass](#) [▶ 475] mit den Daten zur Abbruchursache.

Rückgabeparameter: Eigener Schnittstellenzeiger vom Typ [I_AsyncServiceResultClass](#) [▶ 387].

Beispiel (Auszug)

```

METHOD MyServiceReq : BOOL
VAR_INPUT
    tTimeout : TIME;
END_VAR

```

```

VAR_OUTPUT
    ipResult : I_AsyncServiceResultClass := _fbResult;
END_VAR

IF ipResult.IsBusy() THEN(* service allready running => abort *)
    _state := 0;
    ipResult.SetAbortedEx( stReason := ERROR_INVALID_STATE );
    RETURN;
END_IF
IF tTimeout = T#0S THEN(* invalid parameter => failure *)
    ipResult.SetFailedEx( stError := ERROR_INVALID_VALUE );
    RETURN;
END_IF
ipResult.SetBusy();
_fbTimer( IN := FALSE, PT := tTimeout );
_state := 1;
THIS^();
MyServiceReq := TRUE;

```

6.5.4.12 SetBusy

Setzt den Status in der eigenen Objektinstanz auf „Beschäftigt“.

```
METHOD SetBusy : I_AsyncServiceResultClass
```

Rückgabeparameter: Eigener Schnittstellenzeiger vom Typ [I_AsyncServiceResultClass](#) [► 387].

Beispiel (Auszug)

```

METHOD MyServiceReq : BOOL
VAR_INPUT
    tTimeout : TIME;
END_VAR
VAR_OUTPUT
    ipResult : I_AsyncServiceResultClass := _fbResult;
END_VAR

IF ipResult.IsBusy() THEN(* service allready running => abort *)
    _state := 0;
    ipResult.SetAbortedEx( stReason := ERROR_INVALID_STATE );
    RETURN;
END_IF
IF tTimeout = T#0S THEN(* invalid parameter => failure *)
    ipResult.SetFailedEx( stError := ERROR_INVALID_VALUE );
    RETURN;
END_IF
ipResult.SetBusy();
_fbTimer( IN := FALSE, PT := tTimeout );
_state := 1;
THIS^();
MyServiceReq := TRUE;

```

6.5.4.13 SetCompleted

Setzt den Status in der eigenen Objektinstanz auf „Erledigt“ (ohne Fehler).

```
METHOD SetCompleted : I_AsyncServiceResultClass
```

Rückgabeparameter: Eigener Schnittstellenzeiger vom Typ [I_AsyncServiceResultClass](#) [► 387].

6.5.4.14 SetFailed

Setzt den Fehlerstatus und speichert die Daten zur Fehlerursache in der eigenen Objektinstanz.

```

METHOD SetFailed : I_AsyncServiceResultClass
VAR_INPUT
    ipError : I_ServiceErrorClass;
END_VAR

```

ipError: Schnittstellenzeiger vom Typ [I_ServiceErrorClass](#) [► 392] von einer Objektinstanz mit den Daten zur Fehlerursache.

Rückgabeparameter: Eigener Schnittstellenzeiger vom Typ [I_AsyncServiceResultClass](#) [► 387].

6.5.4.15 SetFailedEx

Setzt den Fehlerstatus und speichert die Daten zur Fehlerursache in der eigenen Objektinstanz.

```
METHOD SetFailedEx : I_AsyncServiceResultClass
VAR_INPUT
    stError : ST_ServiceErrorClass;
END_VAR
```

stError: Strukturierte Variable vom Typ [ST_ServiceErrorClass](#) [► 475] mit den Daten zur Fehlerursache.

Rückgabeparameter: Eigener Schnittstellenzeiger vom Typ [I_AsyncServiceResultClass](#) [► 387].

Beispiel (Auszug)

```
METHOD MyServiceReq : BOOL
VAR_INPUT
    tTimeout : TIME;
END_VAR
VAR_OUTPUT
    ipResult : I_AsyncServiceResultClass := _fbResult;
END_VAR

IF ipResult.IsBusy() THEN(* service allready running => abort *)
    _state := 0;
    ipResult.SetAbortedEx( stReason := ERROR_INVALID_STATE );
    RETURN;
END_IF

IF tTimeout = T#0S THEN(* invalid parameter => failure *)
    ipResult.SetFailedEx( stError := ERROR_INVALID_VALUE );
    RETURN;
END_IF

ipResult.SetBusy();
_fbTimer( IN := FALSE, PT := tTimeout );
_state := 1;
THIS^();
MyServiceReq := TRUE;
```

6.5.4.16 eState

Asynchroner Dienstatus.

```
PROPERTY eState : E_AsyncServiceState {GET}
```

eState: Status vom Typ [E_AsyncServiceState](#) [► 444].

6.5.5 I_ServiceErrorClass

Generische Schnittstelle zur Fehlerabfrage und Fehlerspeicherung. Bei Objekten, die diese Schnittstelle implementieren, können Fehlerinformationen wie z. B. der Fehlercode, Fehlerquelle oder Fehlertext abgefragt oder verglichen werden.

Vererbungshierarchie

I_ServiceErrorClass

Namensraum: [Tc3_Collections](#) [► 96]

Bibliothek: [Tc3_Collections](#) (Tc3_Collections.compiled-library)

```
INTERFACE I_ServiceErrorClass
```

 Methoden

| Name | Beschreibung |
|--|--|
| CopyFrom [▶ 393] | Kopiert Objektdaten von einer anderen Objektinstanz |
| Dump [▶ 393] | Konvertiert eigene Objektdaten in einen formatierten String (z.B. für Logausgaben) |
| Equal [▶ 393] | Wertevergleich der eigenen Objektdaten (Fehlerinformation) mit externen Datenwerten |
| EqualEx [▶ 394] | Wertevergleich der eigenen Objektdaten (Fehlerinformation) mit externen Datenwerten |
| EqualTo [▶ 394] | Wertevergleich der eigenen Objektdaten (Fehlerinformation) mit den Daten einer anderen Objektinstanz |
| IsFailed [▶ 394] | Prüft eigene Objektdaten (Fehlerinformation) auf Misserfolg |
| IsSucceeded [▶ 394] | Prüft eigene Objektdaten (Fehlerinformation) auf Erfolg |
| SetFailedEx [▶ 394] | Speichert Fehlerinformation in der eigenen Objektinstanz |
| SetSucceeded [▶ 395] | Speichert Erfolgsinformation in der eigenen Objektinstanz |

 Eigenschaften

| Name | Typ | Zugriff | Beschreibung |
|---|--|---------|------------------------------------|
| stError [▶ 395] | ST_ServiceErrorClass [▶ 475] | Get | Objektdaten als strukturierter Typ |

6.5.5.1 CopyFrom

Kopiert Objektdaten von einer anderen Objektinstanz.

```
METHOD CopyFrom : BOOL
VAR_INPUT
    ipOther : I_ServiceErrorClass;
END_VAR
```

ipOther: Schnittstellenzeiger vom Typ [I_ServiceErrorClass](#) [[▶ 392](#)] von der Objektinstanz der Datenquelle.

Rückgabeparameter: TRUE => Erfolg, FALSE => Fehler (z. B.: Schnittstellenzeiger ist Null).

6.5.5.2 Dump

Konvertiert eigene Objektdaten in einen formatierten String (z. B. für Logausgaben).

```
METHOD Dump : T_MaxString
```

Rückgabeparameter: String mit dem folgenden Format: '{Severity: [Verbose][Info][Warning][Error][Critical], Id: n[nnnnn], Class: „NameOrGuidId“ }'.

6.5.5.3 Equal

Wertevergleich der eigenen Objektdaten (Fehlerinformation) mit externen Datenwerten.

```
METHOD Equal : BOOL
VAR_INPUT
    uuidOther : GUID;
    nOther : UDINT;
    eOther : TcEventSeverity;
END_VAR
```

uuidOther: Die zu vergleichende Ereignissklasse vom Typ GUID.

nOther: Die zu vergleichende Ereignis-ID.

eOther: Die zu vergleichende Ereignisart vom Typ [TcEventSeverity](#) [[▶ 453](#)].

Rückgabeparameter: TRUE => Daten sind gleich, FALSE => Daten sind unterschiedlich.

6.5.5.4 EqualEx

Wertevergleich der eigenen Objektdaten (Fehlerinformation) mit externen Datenwerten.

```
METHOD EqualEx : BOOL
VAR_INPUT
    stOther : ST_ServiceErrorClass;
END_VAR
```

stOther: Strukturierte Variable vom Typ [ST_ServiceErrorClass](#) [▶ 475] mit den zu vergleichenden Datenwerten.

Rückgabeparameter: TRUE => Daten sind gleich, FALSE => Daten sind unterschiedlich.

6.5.5.5 EqualTo

Wertevergleich der eigenen Objektdaten (Fehlerinformation) mit den Daten einer anderen Objektinstanz.

```
METHOD EqualTo : BOOL
VAR_INPUT
    ipOther : I_ServiceErrorClass;
END_VAR
```

ipOther: Schnittstellenzeiger vom Typ [I_ServiceErrorClass](#) [▶ 392] von der Objektinstanz, deren Daten verglichen werden sollen.

Rückgabeparameter: TRUE => Daten sind gleich, FALSE => Daten sind unterschiedlich.

6.5.5.6 IsFailed

Prüft eigene Objektdaten (Fehlerinformation) auf Misserfolg.

```
METHOD IsFailed : BOOL
```

Rückgabeparameter: TRUE => Misserfolg, FALSE => Erfolg.

6.5.5.7 IsSucceeded

Prüft eigene Objektdaten (Fehlerinformation) auf Erfolg.

```
METHOD IsSucceeded : BOOL
```

Rückgabeparameter: TRUE => Erfolg, FALSE => Misserfolg.

6.5.5.8 SetFailedEx

Speichert Fehlerinformationen in der eigenen Objektinstanz.

```
METHOD SetFailedEx : I_ServiceErrorClass
VAR_INPUT
    stError : ST_ServiceErrorClass;
END_VAR
```

stError: Strukturierte Variable vom Typ [ST_ServiceErrorClass](#) [▶ 475] mit den Daten zur Fehlerursache.

Rückgabeparameter: Eigener Schnittstellenzeiger vom Typ [I_ServiceErrorClass](#) [▶ 392].

Beispiel (Auszug)

```
METHOD DoSomething : BOOL
VAR_INPUT
    sName : STRING;
END_VAR
VAR_OUTPUT
    ipError : I_ServiceErrorClass := _fbError;
END_VAR
IF sName = '' THEN
```

```

    ipError.SetFailedEx( stError := TC_EVENTS.TcGeneralAdsEventClass.InvalidParam );
    RETURN;
END_IF

(* ToDo: *)

ipError.SetSucceeded();
DoSomething := TRUE;

```

6.5.5.9 SetSucceeded

Speichert Erfolgswerte in der eigenen Objektinstanz.

METHOD SetSucceeded : I_ServiceErrorClass

Rückgabeparameter: Eigener Schnittstellenzeiger vom Typ [I_ServiceErrorClass](#) [▶ 392].

Beispiel (Auszug)

```

METHOD DoSomething : BOOL
VAR_INPUT
    sName : STRING;
END_VAR
VAR_OUTPUT
    ipError : I_ServiceErrorClass := _fbError;
END_VAR
IF sName = '' THEN
    ipError.SetFailedEx( stError := TC_EVENTS.TcGeneralAdsEventClass.InvalidParam );
    RETURN;
END_IF

(* ToDo: *)

ipError.SetSucceeded();
DoSomething := TRUE;

```

6.5.5.10 stError

PROPERTY stError : ST_ServiceErrorClass (GET)

stError: Fehlerinformationen als strukturierter Typ [ST_ServiceErrorClass](#) [▶ 475].

6.5.6 I_BaseStreamClass

Vererbungshierarchie

I_BaseStreamClass

INTERFACE I_BaseStreamClass

6.6 Datentypen

6.6.1 E_AcsiAccessPermission

Konfiguriert die Beschränkung des Zugriffs auf einen Datenwert.

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiAccessPermission :
(
    RO := 1,
    WO := 2,

```

```

    RW := 3
) SINT;
END_TYPE

```

Werte

| Name | Beschreibung |
|------|--------------------------|
| RO | Nur Lesezugriff. |
| WO | Nur Schreibzugriff. |
| RW | Lese und Schreibzugriff. |

6.6.2 E_AcsiAdjSt

Namensraum: [Tc3_Acsi](#) [► 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiAdjSt :
(
    Completed      := 1,
    Cancelled      := 2,
    NewAdjustments := 3,
    UnderWay       := 4
) SINT;
END_TYPE

```

Werte

| Name | Beschreibung |
|----------------|--------------|
| Completed | |
| Cancelled | |
| NewAdjustments | |
| UnderWay | |

6.6.3 E_AcsiAlmLev

Namensraum: [Tc3_Acsi](#) [► 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiAlmLev :
(
    Low      := 1,
    Normal  := 2,
    Urgent   := 3
) SINT;
END_TYPE

```

Werte

| Name | Beschreibung |
|--------|--------------|
| Low | |
| Normal | |
| Urgent | |

6.6.4 E_AcsiAnalogueValuePresCond

Konfiguriert die Präsenz der AnalogueValue „i“ und „f“ Attribute in den Statuswerten (mxVal) und Kontrollwerten (ctlVal).

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiAnalogueValuePresCond :
(
    i           := 2#0101,
    f           := 2#1010,
    both_CO_i   := 2#0111,
    both_CO_f   := 3#1011
) SINT;
END_TYPE
```

Werte

| Name | Statuswerte (mxVal) | Kontrollwerte (ctlVal) |
|-----------|---------------------|------------------------|
| i | i | i |
| f | f | f |
| both_CO_i | i und f | i |
| both_CO_f | i und f | f |

6.6.5 E_AcsiAngRef

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiAngRef :
(
    Va           := 0,
    Vb           := 1,
    Vc           := 2,
    Aa           := 3,
    Ab           := 4,
    Ac           := 5,
    Vab          := 6,
    Vbc          := 7,
    Vca          := 8,
    Vother       := 9,
    Aother       := 10,
    Synchrophasor := 11
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|----------------|--------------|
| Va | |
| Vb | |
| Vc | |
| Aa | |
| Ab | |
| Ac | |
| Vab | |
| Vbc | |
| Vca | |
| Vother | |
| Aother | |
| Synchronphasor | |

6.6.6 E_AcsiApplAddCause

Zusätzliche Diagnoseinformation über die Ursache einer negativen Rückmeldung bei der Ausführung eines Dienstes für die Steuerung (Operate, Select, Cancel, usw).

Namensraum: [Tc3_Acsi](#) [► 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiApplAddCause :
(
    Unknown                := 0,
    NotSupported            := 1,
    BlockedBySwitchingHierarchy := 2,
    SelectFailed            := 3,
    InvalidPosition         := 4,
    PositionReached         := 5,
    ParameterChangeInExecution := 6,
    StepLimit               := 7,
    BlockedByMode           := 8,
    BlockedByProcess        := 9,
    BlockedByInterlocking   := 10,
    BlockedBySynchrocheck   := 11,
    CommandAlreadyInExecution := 12,
    BlockedByHealth         := 13,
    OneOfNControl           := 14,
    AbortionByCancel        := 15,
    TimeLimitOver           := 16,
    AbortionByTrip          := 17,
    ObjectNotSelected       := 18,
    ObjectAlreadySelected   := 19,
    NoAccessAuthority        := 20,
    EndedWithOvershoot      := 21,
    AbortionDueToDeviation  := 22,
    AbortionByCommunicationLoss := 23,
    BlockedByCommand        := 24,
    None                     := 25,
    InconsistentParameters  := 26,
    LockedByOtherClient      := 27
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|-----------------------------|---------------------------------|
| Unknown | Unknown. |
| NotSupported | Not-supported. |
| BlockedBySwitchingHierarchy | Blocked-by-switching-hierarchy. |
| SelectFailed | Select-failed. |
| InvalidPosition | Invalid-position. |
| PositionReached | Position-reached. |
| ParameterChangeInExecution | Parameter-change-in-execution. |
| StepLimit | Step-limit. |
| BlockedByMode | Blocked-by-Mode. |
| BlockedByProcess | Blocked-by-process. |
| BlockedByInterlocking | Blocked-by-interlocking. |
| BlockedBySynchrocheck | Blocked-by-synchrocheck. |
| CommandAlreadyInExecution | Command-already-in-execution. |
| BlockedByHealth | Blocked-by-health. |
| OneOfNControl | 1-of-n-control. |
| AbortionByCancel | Abortion-by-cancel. |
| TimeLimitOver | Time-limit-over. |
| AbortionByTrip | Abortion-by-trip. |
| ObjectNotSelected | Object-not-selected. |
| ObjectAlreadySelected | Object-already-selected. |
| NoAccessAuthority | No-access-authority. |
| EndedWithOvershoot | Ended-with-overshoot. |
| AbortionDueToDeviation | Abortion-due-to-deviation. |
| AbortionByCommunicationLoss | Abortion-by-communication-loss. |
| lockedByCommand | Blocked-by-command. |
| None | None. |
| InconsistentParameters | Inconsistent-parameters. |
| LockedByOtherClient | Locked-by-other-client. |

6.6.7 E_AcsiApplError

Diagnoseinformation über die Fehlerquelle (z. B. die Zustandsmaschine des Dienstes, der Operate- oder TimeActivated-Dienst selbst oder andere Quelle).

Namensraum: Tc3_Acsi [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiApplError :
(
  NoError           := 0,
  Unknown           := 1,
  TimeoutTestNotOk := 2,
  OperatorTestNotOk := 3
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|-------------------|-----------------------|
| NoError | No error. |
| Unknown | Unknown. |
| TimeoutTestNotOk | Timeout test not ok. |
| OperatorTestNotOk | Operator test not ok. |

6.6.8 E_AcsiAutoRecSt

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiAutoRecSt :
(
  Ready           := 1,
  InProgress      := 2,
  Successful       := 3,
  WaitingForTrip  := 4,
  TripIssuedByProtection := 5,
  FaultDisappeared := 6,
  WaitToComplete := 7,
  CircuitBreakerClosed := 8,
  CycleUnsuccessful := 9,
  Unsuccessful    := 10,
  Aborted         := 11
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|------------------------|--------------|
| Ready | |
| InProgress | |
| Successful | |
| WaitingForTrip | |
| TripIssuedByProtection | |
| FaultDisappeared | |
| WaitToComplete | |
| CircuitBreakerClosed | |
| CycleUnsuccessful | |
| Unsuccessful | |
| Aborted | |

6.6.9 E_AcsiBeh

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiBeh :
(
  On           := 1,
  OnBlocked    := 2,
  Test         := 3,
  TestBlocked  := 4,

```

```
Off := 5
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|-------------|---|
| On | |
| OnBlocked | "blocked" in Ed1 is changed in Ed2 to "on-blocked". |
| Test | |
| TestBlocked | |
| Off | |

6.6.10 E_AcsiCBOpCap

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiCBOpCap :
(
  None := 1,
  Open := 2,
  CloseOpen := 3,
  OpenCloseOpen := 4,
  CloseOpenCloseOpen := 5,
  OpenCloseOpenCloseOpen := 6,
  More := 7
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|------------------------|--------------|
| None | |
| Open | |
| CloseOpen | |
| OpenCloseOpen | |
| CloseOpenCloseOpen | |
| OpenCloseOpenCloseOpen | |
| More | |

6.6.11 E_AcsiCheck

Prüfungen, die bei der Ausführung eines Dienstes für die Steuerung durchgeführt werden sollen. Der Aufzählungswert wird in 2 Bits codiert. Die restlichen 6 Bits werden nicht verwendet.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiCheck :
(
  NoCheck := 2#00,
  SynchroCheck := 2#01,
  InterlockingCheck := 2#10,
  Both := 2#11
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|-------------------|---------------------|
| NoCheck | Keine Prüfung. |
| SynchroCheck | Synchro-check. |
| InterlockingCheck | Interlocking-check. |
| Both | Beide Prüfungen. |

6.6.12 E_AcsiClcIntvTyp

Namensraum: [Tc3_Acsi](#) [[▶](#) [96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiClcIntvTyp :
(
  Ms      := 1,
  PerCycle := 2,
  Cycle   := 3,
  Day     := 4,
  Week    := 5,
  Month   := 6,
  Year    := 7,
  External := 8
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|----------|--------------|
| Ms | |
| PerCycle | |
| Cycle | |
| Day | |
| Week | |
| Month | |
| Year | |
| External | |

6.6.13 E_AcsiClcMod

Namensraum: [Tc3_Acsi](#) [[▶](#) [96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiClcMod :
(
  Total    := 1,
  Period   := 2,
  Sliding  := 3
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|---------|--------------|
| Total | |
| Period | |
| Sliding | |

6.6.14 E_AcsiClcMth

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiClcMth :
(
  Unspecified      := 1,
  TrueRms          := 2,
  PeakFundamental := 3,
  RmsFundamental  := 4,
  Min_             := 5,
  Max_             := 6,
  Avg              := 7,
  Sdv              := 8,
  Prediction       := 9,
  Rate             := 10
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|-----------------|--------------|
| Unspecified | |
| TrueRms | |
| PeakFundamental | |
| RmsFundamental | |
| Min_ | |
| Max_ | |
| Avg | |
| Sdv | |
| Prediction | |
| Rate | |

6.6.15 E_AcsiClcRfTyp

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiClcRfTyp :
(
  Ms           := 1,
  PerCycle    := 2,
  Cycle       := 3,
  Day         := 4,
  Week        := 5,
  Month       := 6,
  Year        := 7,

```

```

    External := 8
) SINT;
END_TYPE

```

Werte

| Name | Beschreibung |
|----------|--------------|
| Ms | |
| PerCycle | |
| Cycle | |
| Day | |
| Week | |
| Month | |
| Year | |
| External | |

6.6.16 E_AcsiClcTotVA

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiClcTotVA :
(
    Vector      := 1,
    Arithmetic := 2
) SINT;
END_TYPE

```

Werte

| Name | Beschreibung |
|------------|--------------|
| Vector | |
| Arithmetic | |

6.6.17 E_AcsiCmdQual

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiCmdQual :
(
    Pulse       := 0,
    Persistent_ := 1
) SINT;
END_TYPE

```

Werte

| Name | Beschreibung |
|-------------|--------------|
| Pulse | |
| Persistent_ | |

6.6.18 E_AcsiCtlModels

Kontrollmodell für die Schaltersteuerung.

Namensraum: [Tc3_Acsi](#) [[▶](#) [96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiCtlModels :
(
    StatusOnly           := 0,
    DirectWithNormalSecurity := 1,
    SBOWithNormalSecurity := 2,
    DirectWithEnhancedSecurity := 3,
    SBOWithEnhancedSecurity := 4
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|----------------------------|---|
| StatusOnly | Nur Statusinformation, keine Kontrolle erlaubt. |
| DirectWithNormalSecurity | Direct-with-normal-security (direct-operate). |
| SBOWithNormalSecurity | SBO-with-normal-security (operate-once operate-many). |
| DirectWithEnhancedSecurity | Direct-with-enhanced-security (direct-operate). |
| SBOWithEnhancedSecurity | SBO-with-enhanced-security (operate-once operate-many). |

6.6.19 E_AcsiCtlReport

Kontrollbefehle einer Berichtssteuerblockinstanz für gepufferte oder ungepufferte Reports (BRCB, URCB).

Namensraum: [Tc3_Acsi](#) [[▶](#) [96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiCtlReport :
(
    None      := 0,
    Enable   := 1,
    GI       := 2,
    Disable  := 3
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|---------|--|
| None | Kein Befehl (Ruhezustand). |
| Enable | Befehl zum Aktivieren der Reports (gepufferten oder ungepufferten). |
| GI | Befehl zum Aktivieren der Generalabfrage. |
| Disable | Befehl zum Deaktivieren der Reports (gepufferten oder ungepufferten) und der Generalabfrage. |

6.6.20 E_AcsiCtlService

Kontrollbefehle für die Dienste der Datenobjekte zur Schaltersteuerung.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiCtlService :
(
  None           := 0,
  Select         := 1,
  Operate        := 2,
  SelectAndOperate := 3,
  Cancel         := 4
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|------------------|---|
| None | Kein Befehl (Ruhezustand). |
| Select | Aktiviert den Select- oder SelectWithValue-Befehl. |
| Operate | Aktiviert den Operate-Befehl. |
| SelectAndOperate | Aktiviert den Select-Befehl gefolgt von dem Operate-Befehl. |
| Cancel | Aktiviert den Cancel-Befehl. |

6.6.21 E_AcsiCycTrMod

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiCycTrMod :
(
  ThreePhaseTripping := 1,
  OneOrThreePhaseTripping := 2,
  Specific             := 3
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|-------------------------|--------------|
| ThreePhaseTripping | |
| OneOrThreePhaseTripping | |
| Specific | |

6.6.22 E_AcsiDbpos

Doppelpunkt-Statuswert. Der Aufzählungswert wird in 2 Bits codiert. Die restlichen 6 Bits werden nicht verwendet.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiDbpos :
(
  Intermediate := 2#00,
```

```

Off      := 2#01,
On       := 2#10,
Bad      := 2#11
) SINT;
END_TYPE

```

Werte

| Name | Beschreibung |
|--------------|-----------------|
| Intermediate | Zwischenzustand |
| Off | Aus |
| On | An |
| Bad | Ungut |

6.6.23 E_AcsiDir

Namensraum: [Tc3_Acsi](#) [[▶](#) [96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiDir :
(
    Unknown := 0,
    Forward := 1,
    Backward := 2,
    Both := 3
) SINT;
END_TYPE

```

Werte

| Name | Beschreibung |
|----------|--------------|
| Unknown | |
| Forward | |
| Backward | |
| Both | |

6.6.24 E_AcsiDirMod

Namensraum: [Tc3_Acsi](#) [[▶](#) [96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiDirMod :
(
    NonDirectional := 1,
    Forward := 2,
    Reverse := 3
) SINT;
END_TYPE

```

Werte

| Name | Beschreibung |
|----------------|--------------|
| NonDirectional | |
| Forward | |
| Reverse | |

6.6.25 E_AcsiEEHealth

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiEEHealth :
(
    Ok      := 1,
    Warning := 2,
    Alarm   := 3
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|---------|--------------|
| Ok | |
| Warning | |
| Alarm | |

6.6.26 E_AcsiFailMod

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiFailMod :
(
    Current           := 1,
    BreakerStatus     := 2,
    BothCurrentAndBreakerStatus := 3,
    Other              := 4
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|-----------------------------|--------------|
| Current | |
| BreakerStatus | |
| BothCurrentAndBreakerStatus | |
| Other | |

6.6.27 E_AcsiFanCtl

Kontrollwert für Lüftersteuerung.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiFanCtl :
(
    Inactive := 1,
    Stage1   := 2,

```

```

    Stage2 := 3,
    Stage3 := 4
) SINT;
END_TYPE

```

Werte

| Name | Beschreibung |
|----------|--------------|
| Inactive | |
| Stage1 | |
| Stage2 | |
| Stage3 | |

6.6.28 E_AcsiFanCtlGen

Generischer Kontrollwert für Lüftersteuerung.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiFanCtlGen :
(
    Inactive := 1,
    Stage1   := 2,
    Stage2   := 3,
    Stage3   := 4
) SINT;
END_TYPE

```

Werte

| Name | Beschreibung |
|----------|--------------|
| Inactive | |
| Stage1 | |
| Stage2 | |
| Stage3 | |

6.6.29 E_AcsiFc

Funktionale Gruppe als Aufzählungstyp.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiFc :
(
    None := 16#000000,
    MX   := 16#000001,
    ST_  := 16#000002,
    CO   := 16#000004,
    CF   := 16#000008,
    DC   := 16#000010,
    SP   := 16#000020,
    SG   := 16#000040,
    RP   := 16#000080,

    LG   := 16#000100,
    BR   := 16#000200,

```

```

GO := 16#000400,
GS := 16#000800,
SV := 16#001000,
SE := 16#002000,
MS := 16#004000,
SC := 16#008000,

US := 16#010000,
EX := 16#020000,

XX := 16#03FFFF
)DWORD;
END_TYPE

```

Werte

| Name | Beschreibung |
|------|--------------|
| None | |
| MX | |
| ST_ | |
| CO | |
| CF | |
| DC | |
| SP | |
| SG | |
| RP | |
| LG | |
| BR | |
| GO | |
| GS | |
| SV | |
| SE | |
| MS | |
| SC | |
| US | |
| EX | |
| XX | |

6.6.30 E_AcsiFilTyp

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiFilTyp :
(
  LowPass := 1,
  HighPass := 2,
  Bandpass := 3,
  Bandstop := 4,
  Deadband := 5
)SINT;
END_TYPE

```

Werte

| Name | Beschreibung |
|----------|--------------|
| LowPass | |
| HighPass | |
| Bandpass | |
| Bandstop | |
| Deadband | |

6.6.31 E_AcsiFltLoop

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiFltLoop :
(
  PhaseAToGround := 1,
  PhaseBToGround := 2,
  PhaseCToGround := 3,
  PhaseAToB      := 4,
  PhaseBToC      := 5,
  PhaseCToA      := 6,
  Other          := 7
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|----------------|--------------|
| PhaseAToGround | |
| PhaseBToGround | |
| PhaseCToGround | |
| PhaseAToB | |
| PhaseBToC | |
| PhaseCToA | |
| Other | |

6.6.32 E_AcsiGnSt

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiGnSt :
(
  Stopped   := 1,
  Stopping  := 2,
  Started   := 3,
  Starting  := 4,
  Disabled  := 5
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|----------|--------------|
| Stopped | |
| Stopping | |
| Started | |
| Starting | |
| Disabled | |

6.6.33 E_AcsiHealth

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiHealth :
(
    Ok      := 1,
    Warning := 2,
    Alarm   := 3
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|---------|--------------|
| Ok | |
| Warning | |
| Alarm | |

6.6.34 E_AcsiHvRef

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiHvRef :
(
    Fundamental := 0,
    Rms          := 1,
    Absolute     := 2
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|-------------|--------------|
| Fundamental | |
| Rms | |
| Absolute | |

6.6.35 E_AcsiIntrDetMth

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiIntrDetMth :
(
    Voltage := 1,
    VoltageAndCurrent := 2,
    VoltageAndNormallyOpenBreakerContact := 3,
    VoltageAndNormallyClosedBreakerContact := 4,
    VoltageAndBothNormallyOpenAndNormallyClosedBreakerContacts := 5,
    NormallyOpenBreakerContact := 6,
    NormallyClosedBreakerContacts := 7,
    BothNormallyOpenAndNormallyClosedBreakerContacts := 8
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|--|--------------|
| Voltage | |
| VoltageAndCurrent | |
| VoltageAndNormallyOpenBreakerContact | |
| VoltageAndNormallyClosedBreakerContact | |
| VoltageAndBothNormallyOpenAndNormallyClosedBreakerContacts | |
| NormallyOpenBreakerContact | |
| NormallyClosedBreakerContacts | |
| BothNormallyOpenAndNormallyClosedBreakerContacts | |

6.6.36 E_AcsiLevMod

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiLevMod :
(
    PositiveOrRising := 1,
    NegativeOrFalling := 2,
    Both := 3,
    Other := 4
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|-------------------|--------------|
| PositiveOrRising | |
| NegativeOrFalling | |
| Both | |
| Other | |

6.6.37 E_AcsiLivDeaMod

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiLivDeaMod :
```

```
(
  DeadLineDeadBus           := 1,
  LiveLineDeadBus          := 2,
  DeadLineLiveBus          := 3,
  DeadLineDeadBusOrLiveLineDeadBus := 4,
  DeadLineDeadBusOrDeadLineLiveBus := 5,
  LiveLineDeadBusOrDeadLineLiveBus := 6,
  DeadLineDeadBusOrLiveLineDeadBusOrDeadLineLiveBus := 7
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|---|--------------|
| DeadLineDeadBus | |
| LiveLineDeadBus | |
| DeadLineLiveBus | |
| DeadLineDeadBusOrLiveLineDeadBus | |
| DeadLineDeadBusOrDeadLineLiveBus | |
| LiveLineDeadBusOrDeadLineLiveBus | |
| DeadLineDeadBusOrLiveLineDeadBusOrDeadLineLiveBus | |

6.6.38 E_AcsiLogicalNodeClass

Objektklassen eines logischen Knoten.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiLogicalNodeClass :
(
  DataObject           := 1,
  DataSet              := 2,
  BufferedReportControlBlock := 3,
  UnbufferedReporControlBlock := 4,
  LogControlBlock      := 5,
  SettingGroupControlBlock := 6,
  Log_                  := 7,
  GooseControlBlock    := 8,
  GsseControlBlock     := 9,
  MulticastSampledValueControlBlock := 10,
  UnicastSampledValueControlBlock := 11
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|-----------------------------------|--------------|
| DataObject | |
| DataSet | |
| BufferedReportControlBlock | |
| UnbufferedReporControlBlock | |
| LogControlBlock | |
| SettingGroupControlBlock | |
| Log_ | |
| GooseControlBlock | |
| GsseControlBlock | |
| MulticastSampledValueControlBlock | |
| UnicastSampledValueControlBlock | |

6.6.39 E_AcsiMechHealth

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiMechHealth :
(
    Ok      := 1,
    Warning := 2,
    Alarm   := 3
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|---------|--------------|
| Ok | |
| Warning | |
| Alarm | |

6.6.40 E_AcsiMod

Dieser Parameter kontrolliert das Verhalten des Logischen Knoten (z.B. bei der Verarbeitung der empfangenen Daten).

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiMod :
(
    On           := 1,
    OnBlocked    := 2,
    Test         := 3,
    TestBlocked  := 4,
    Off          := 5
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|-------------|--------------|
| On | |
| OnBlocked | |
| Test | |
| TestBlocked | |
| Off | |

6.6.41 E_AcsiMonth

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiMonth :
(
{ attribute 'hide'}    reserved := 0,
  January      := 1,
  February     := 2,
  March        := 3,
  April        := 4,
  May          := 5,
  June         := 6,
  July         := 7,
  August       := 8,
  September    := 9,
  October      := 10,
  November     := 11,
  December    := 12
) SINT;
END_TYPE
```

6.6.42 E_AcsiMultiplier

Multiplikator.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiMultiplier :
(
  Yocto := -24,
  Zepto := -21,
  Atto  := -18,
  Femto := -15,
  Pico  := -12,
  Nano  := -9,
  Micro := -6,
  Milli := -3,
  Centi := -2,
  Deci  := -1,
  None  := 0,
  Deca  := 1,
  Hecto := 2,
  Kilo  := 3,
  Mega  := 6,
  Giga  := 9,
  Tera  := 12,
  Petra := 15,
  Exa   := 18,
  Zetta := 21,
  Yotta := 24
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|-------|--|
| Yocto | Value: 10 ⁽⁻²⁴⁾ , Symbol: y |
| Zepto | Value: 10 ⁽⁻²¹⁾ , Symbol: z |
| Atto | Value: 10 ⁽⁻¹⁸⁾ , Symbol: a |
| Femto | Value: 10 ⁽⁻¹⁵⁾ , Symbol: f |
| Pico | Value: 10 ⁽⁻¹²⁾ , Symbol: p |
| Nano | Value: 10 ⁽⁻⁹⁾ , Symbol: n |
| Micro | Value: 10 ⁽⁻⁶⁾ , Symbol: u |
| Milli | Value: 10 ⁽⁻³⁾ , Symbol: m |
| Centi | Value: 10 ⁽⁻²⁾ , Symbol: c |
| Deci | Value: 10 ⁽⁻¹⁾ , Symbol: d |
| None | Value: 1 |
| Deca | Value: 10 ⁽¹⁾ , Symbol: da |
| Hecto | Value: 10 ⁽²⁾ , Symbol: h |
| Kilo | Value: 10 ⁽³⁾ , Symbol: k |
| Mega | Value: 10 ⁽⁶⁾ , Symbol: M |
| Giga | Value: 10 ⁽⁹⁾ , Symbol: G |
| Tera | Value: 10 ⁽¹²⁾ , Symbol: T |
| Petra | Value: 10 ⁽¹⁵⁾ , Symbol: P |
| Exa | Value: 10 ⁽¹⁸⁾ , Symbol: E |
| Zetta | Value: 10 ⁽²¹⁾ , Symbol: Z |
| Yotta | Value: 10 ⁽²⁴⁾ , Symbol: Y |

6.6.43 E_AcsiOccPer

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiOccPer :
(
    Hour := 0,
    Day := 1,
    Week := 2,
    Month := 3,
    Year := 4
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|-------|--------------|
| Hour | Stunde |
| Day | Tag |
| Week | Woche |
| Month | Monat |
| Year | Jahr |

6.6.44 E_AcsiOccType

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiOccType :
(
  Time_      := 0,
  WeekDay    := 1,
  WeekOfYear := 2,
  DayOfMonth := 3,
  DayOfYear  := 4
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|------------|--------------|
| Time_ | |
| WeekDay | |
| WeekOfYear | |
| DayOfMonth | |
| DayOfYear | |

6.6.45 E_AcsiOpModRect

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiOpModRect :
(
  CurrentControl    := 1,
  VoltageControl    := 2,
  ActivePowerControl := 3
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|--------------------|--------------|
| CurrentControl | |
| VoltageControl | |
| ActivePowerControl | |

6.6.46 E_AcsiOpModSyn

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiOpModSyn :
(
  AutomaticSynchronising := 1,
  AutomaticParalleling   := 2,
  Manual                  := 3,
  Test                    := 4
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|------------------------|--------------|
| AutomaticSynchronising | |
| AutomaticParalleling | |
| Manual | |
| Test | |

6.6.47 E_AcsiOrCategory

Befehlsauftraggeber-/Urheber-Kategorie.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiOrCategory :
(
    NotSupported      := 0,
    BayControl        := 1,
    StationControl    := 2,
    RemoteControl     := 3,
    AutomaticBay      := 4,
    AutomaticStation  := 5,
    AutomaticRemote   := 6,
    Maintenance       := 7,
    Process           := 8
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|------------------|--------------|
| NotSupported | |
| BayControl | |
| StationControl | |
| RemoteControl | |
| AutomaticBay | |
| AutomaticStation | |
| AutomaticRemote | |
| Maintenance | |
| Process | |

6.6.48 E_AcsiParColMod

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiParColMod :
(
    MasterSlave := 1,
    MasterSlaveWithFixedSlavePosition := 2,
    MasterSlaveWithVariableSlavePosiiton := 3,
    ParallelOperationWithoutCommunication := 4
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|---------------------------------------|--------------|
| MasterSlave | |
| MasterSlaveWithFixedSlavePosition | |
| MasterSlaveWithVariableSlavePosition | |
| ParallelOperationWithoutCommunication | |

6.6.49 E_AcsiParMod

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiParMod :
(
  Master      := 1,
  Slave       := 2,
  Independent := 3
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|-------------|--------------|
| Master | |
| Slave | |
| Independent | |

6.6.50 E_AcsiParTraMod

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiParTraMod :
(
  NoModePredefined      := 1,
  Master                 := 2,
  Follower               := 3,
  PowerFactor           := 4,
  NegativeReactance     := 5,
  CirculatingCurrent    := 6,
  CirculatingReactiveCurrent := 7,
  CirculatingReactiveCurrentByEqualizingCalculatedTransformerPowerFactor := 8
) SINT;
END_TYPE
```


Werte

| Name | Beschreibung |
|--|--------------|
| NoModePredefined | |
| Master | |
| Follower | |
| PowerFactor | |
| NegativeReactance | |
| CirculatingCurrent | |
| CirculatingReactiveCurrent | |
| CirculatingReactiveCurrentByEqualizingCalculatedTransformerPowerFactor | |

6.6.51 E_AcsiPFSign

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiPFSign :
(
    ActivePower := 1,
    LeadLag     := 2
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|-------------|--------------|
| ActivePower | |
| LeadLag | |

6.6.52 E_AcsiPhsRef

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiPhsRef :
(
    A := 0,
    B := 1,
    C := 2
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|------|--------------|
| A | |
| B | |
| C | |

6.6.53 E_AcsiPhyHealth

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiPhyHealth :
(
    Ok      := 1,
    Warning := 2,
    Alarm   := 3
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|---------|--------------|
| Ok | |
| Warning | |
| Alarm | |

6.6.54 E_AcsiPIDAlg

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiPIDAlg :
(
    P   := 1,
    I   := 2,
    D   := 3,
    PI  := 4,
    PD  := 5,
    ID  := 6,
    PID := 7
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|------|--------------|
| P | |
| I | |
| D | |
| PI | |
| PD | |
| ID | |
| PID | |

6.6.55 E_AcsiPmpCtl

Kontrollwert für Pumpensteuerung.

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiPmpCtl :
(
    Inactive := 1,
    Stage1   := 2,
    Stage2   := 3,
    Stage3   := 4
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|----------|--------------|
| Inactive | |
| Stage1 | |
| Stage2 | |
| Stage3 | |

6.6.56 E_AcsiPmpCtlGen

Generischer Kontrollwert für Pumpensteuerung.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiPmpCtlGen :
(
    Inactive := 1,
    Stage1   := 2,
    Stage2   := 3,
    Stage3   := 4
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|----------|--------------|
| Inactive | |
| Stage1 | |
| Stage2 | |
| Stage3 | |

6.6.57 E_AcsiPolQty

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiPolQty :
(
    None                := 1,
    ZeroSequenceCurrent := 2,
    ZeroSequenceVoltage := 3,
    NegativeSequenceVoltage := 4,
    PhaseToPhaseVoltages := 5,
    PhaseToGroundVoltages := 6,

```

```

    PositiveSequenceVoltage := 7
) SINT;
END_TYPE

```

Werte

| Name | Beschreibung |
|-------------------------|--------------|
| None | |
| ZeroSequenceCurrent | |
| ZeroSequenceVoltage | |
| NegativeSequenceVoltage | |
| PhaseToPhaseVoltages | |
| PhaseToGroundVoltages | |
| PositiveSequenceVoltage | |

6.6.58 E_AcsiPOWCap

Namensraum: [Tc3_Acsi \[► 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiPOWCap :
(
    None           := 1,
    Close          := 2,
    Open           := 3,
    CloseAndOpen  := 4
) SINT;
END_TYPE

```

Werte

| Name | Beschreibung |
|--------------|--------------|
| None | |
| Close | |
| Open | |
| CloseAndOpen | |

6.6.59 E_AcsiQualitySource

Namensraum: [Tc3_Acsi \[► 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiQualitySource :
(
    Process        := 0,
    Substituted    := 1
) SINT;
END_TYPE

```

Werte

| Name | Beschreibung |
|-------------|--------------|
| Process | Prozess |
| Substituted | Ersetzt |

6.6.60 E_AcsiQualityValidity

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiQualityValidity :
(
    Good           := 0,
    Invalid        := 1,
    Reserved       := 2,
    Questionable   := 3
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|--------------|--------------|
| Good | |
| Invalid | |
| Reserved | |
| Questionable | |

6.6.61 E_AcsiRange

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiRange :
(
    Normal        := 0,
    High          := 1,
    Low           := 2,
    HighHigh      := 3,
    LowLow        := 4
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|----------|--------------|
| Normal | |
| High | |
| Low | |
| HighHigh | |
| LowLow | |

6.6.62 E_AcsiRcdMod

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiRcdMod :
```

```
(
  OverwriteExistingValues := 1,
  StopWhenFullOrSaturated := 2
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|-------------------------|--------------|
| OverwriteExistingValues | |
| StopWhenFullOrSaturated | |

6.6.63 E_AcsiReTrMod

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiReTrMod :
(
  Off := 1,
  WithoutCheck := 2,
  WithCurrentCheck := 3,
  WithBreakerStatusCheck := 4,
  WithCurrentAndBreakerStatusCheck := 5,
  OtherChecks := 6
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|----------------------------------|--------------|
| Off | |
| WithoutCheck | |
| WithCurrentCheck | |
| WithBreakerStatusCheck | |
| WithCurrentAndBreakerStatusCheck | |
| OtherChecks | |

6.6.64 E_AcsiRotDir

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiRotDir :
(
  Clockwise := 1,
  CounterClockwise := 2,
  Unknown := 3
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|------------------|--------------|
| Clockwise | |
| CounterClockwise | |
| Unknown | |

6.6.65 E_AcsiRstMod

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiRstMod :
(
  None := 1,
  Harmonic2 := 2,
  Harmonic5 := 3,
  Harmonic2And5 := 4,
  WaveformAnalysis := 5,
  WaveformAnalysisAndHarmonic2 := 6,
  Other := 7,
  WaveformAnalysisAndHarmonic5 := 8,
  WaveformAnalysisAndHarmonic2AndHarmonic5 := 9
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|--|--------------|
| None | |
| Harmonic2 | |
| Harmonic5 | |
| Harmonic2And5 | |
| WaveformAnalysis | |
| WaveformAnalysisAndHarmonic2 | |
| Other | |
| WaveformAnalysisAndHarmonic5 | |
| WaveformAnalysisAndHarmonic2AndHarmonic5 | |

6.6.66 E_AcsiSboClasses

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiSboClasses :
(
  OperateOnce := 0,
  OperateMany := 1
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|-------------|---------------|
| OperateOnce | Operate-once. |
| OperateMany | Operate-many. |

6.6.67 E_AcsiSeqT

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiSeqT :
(
    PosNegZero := 0,
    DirQuadZero := 1
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|-------------|--------------|
| PosNegZero | |
| DirQuadZero | |

6.6.68 E_AcsiServerDirectoryClass

Objektklassen eines Servers.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiServerDirectoryClass :
(
    LogicalDevice := 1,
    File := 2
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|---------------|-----------------|
| LogicalDevice | Logisches Gerät |
| File | Datei |

6.6.69 E_AcsiServiceError

Fehlercodes, die in der Ereignisbehandlungsroutine der Dienstrückmeldung gemeldet werden. Bei der Verwendung des MMS Mappings werden auch die MMS Fehlercodes in diese Fehlercodes umgewandelt (gemappt).

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiServiceError :
(
  NoError := 0,
  InstanceNotAvailable := 1,
  InstanceInUse := 2,
  AccessViolation := 3,
  AccessNotAllowedInCurrentState := 4,
  ParameterValueInappropriate := 5,
  ParameterValueInconsistent := 6,
  ClassNotSupported := 7,
  InstanceLockedByOtherClient := 8,
  ControlMustBeSelected := 9,
  TypeConflict := 10,
  FailedDueToCommunicationsConstraint := 11,
  FailedDueToServerConstraint := 12
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|-------------------------------------|--------------|
| NoError | |
| InstanceNotAvailable | |
| InstanceInUse | |
| AccessViolation | |
| AccessNotAllowedInCurrentState | |
| ParameterValueInappropriate | |
| ParameterValueInconsistent | |
| ClassNotSupported | |
| InstanceLockedByOtherClient | |
| ControlMustBeSelected | |
| TypeConflict | |
| FailedDueToCommunicationsConstraint | |
| FailedDueToServerConstraint | |

6.6.70 E_AcsiServiceType

Diensttyp.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiServiceType :
(
  Unknown := 0,
  Associate := 1,
  Abort := 2,
  Release := 3,
  GetServerDirectory := 4,
  GetLogicalDeviceDirectory := 5,
  GetAllDataValues := 6,
  GetDataValues := 7,
  SetDataValues := 8,
  GetDataDirectory := 9,
  GetDataDefinition := 10,
  GetDataSetValues := 11,
  SetDataSetValues := 12,
  CreateDataSet := 13,
  DeleteDataSet := 14,
  GetDataSetDirectory := 15,

```

```

SelectActiveSG           := 16,
SelectEditSG            := 17,
SetEditSGValue          := 18,
ConfirmEditSGValues     := 19,
GetEditSGValue          := 20,
GetSGCBValues           := 21,
Report                  := 22,
GetBRCBValues           := 23,
SetBRCBValues           := 24,
GetURCBValues           := 25,
SetURCBValues           := 26,
GetLCBValues            := 27,
SetLCBValues            := 28,
QueryLogByTime          := 29,
QueryLogAfter           := 30,
GetLogStatusValues      := 31,
SendGOOSEMessage       := 32,
GetGoCBValues           := 33,
SetGoCBValues           := 34,
GetGoReference          := 35,
GetGOOSEElementNumber  := 36,
SendMSVMessage          := 37,
GetMSVCBValues         := 38,
SetMSVCBValues         := 39,
SendUSVMessage         := 40,
GetUSVCBValues         := 41,
SetUSVCBValues         := 42,
Select                  := 43,
SelectWithValue         := 44,
Cancel                  := 45,
Operate                 := 46,
CommandTermination     := 47,
TimeActivatedOperate   := 48,
GetFile                 := 49,
SetFile                 := 50,
DeleteFile              := 51,
GetFileAttributeValues := 52,
TimeSynchronization    := 53,
InternalChangeUnknown  := 54,
(* TwinCAT specific *)
GetLogicalNodeDirectory := 100,
GetAllServerValues     := 101,
GetAllLogicalDeviceValues := 102,
SetAllDataValues       := 103
) SINT;
END_TYPE

```

6.6.71 E_AcsiSetCharact

Namensraum: [Tc3_Acsi](#) [[↗ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiSetCharact :
(
  AnsiExtremlyInverse := 1,
  AnsiVeryInverse     := 2,
  AnsiNormalInverse   := 3,
  AnsiModerateInverse := 4,
  AnsiDefiniteTime    := 5,
  LongTimeExtermelyInverse := 6,
  LongTimeVeryInverse := 7,
  LongTimeInverse     := 8,
  IecNormalInverse    := 9,
  IecVeryInverse      := 10,
  IecInverse          := 11,
  IecExtremelyInverse := 12,
  IecShortTimeInverse := 13,
  IecLongTimeInverse  := 14,
  IecDefiniteTime     := 15,
  { attribute 'hide'} reserved := 16,
  Polynom1             := 17,
  Polynom2             := 18,
  Polynom3             := 19,

```

```

Polynom4 := 20,
Polynom5 := 21,
Polynom6 := 22,
Polynom7 := 23,
Polynom8 := 24,
Polynom9 := 25,
Polynom10 := 26,
Polynom11 := 27,
Polynom12 := 28,
Polynom13 := 29,
Polynom14 := 30,
Polynom15 := 31,
Polynom16 := 32,
Multiline1 := 33,
Multiline2 := 34,
Multiline3 := 35,
Multiline4 := 36,
Multiline5 := 37,
Multiline6 := 38,
Multiline7 := 39,
Multiline8 := 40,
Multiline9 := 41,
Multiline10 := 42,
Multiline11 := 43,
Multiline12 := 44,
Multiline13 := 45,
Multiline14 := 46,
Multiline15 := 47,
Multiline16 := 48
) SINT;
END_TYPE

```

6.6.72 E_AcsiSev

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiSev :
(
    Unknown := 0,
    Critical := 1,
    Major := 2,
    Minor := 3,
    Warning := 4
) SINT;
END_TYPE

```

Werte

| Name | Beschreibung |
|----------|--------------|
| Unknown | |
| Critical | |
| Major | |
| Minor | |
| Warning | |

6.6.73 E_AcsiShOpCap

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiShOpCap :

```

```
(
  None           := 1,
  Open           := 2,
  Close          := 3,
  OpenAndClose  := 4
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|--------------|--------------|
| None | |
| Open | |
| Close | |
| OpenAndClose | |

6.6.74 E_AcsiSIUnit

SI Einheit.

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiSIUnit :
(
(* Base units: *)
  None           :=1, (* Dimensionless *)
  Meter          :=2, (* Length m *)
  Kilogram       :=3, (* Mass kg *)
  Second         :=4, (* Time s *)
  Ampere         :=5, (* Current A *)
  Kelvin         :=6, (* Temperature K *)
  Mole           :=7, (* Amount of substance mol *)
  Candela       :=8, (* Luminous intensity cd *)

(*Derived units: *)
  Degrees        :=9,  (* Plane angle deg *)
  Radian         :=10, (* Plane angle rad *)
  Steradian      :=11, (* Solid angle sr *)
  Gray           :=21, (* Absorbed dose (J/kg) Gy *)
  Becquerel      :=22, (* Activity (1/s) q *)
  DegreesCelsius :=23, (* Relative temperature °C *)
  Sievert        :=24, (* Dose equivalent (J/kg) Sv *)
  Farad          :=25, (* Electric capacitance (C/V) F *)
  Coulomb        :=26, (* Electric charge (As) C *)
  Siemens        :=27, (* Electric conductance (A/V) S *)
  Henry          :=28, (* Electric inductance (Wb/A) H *)
  Volt           :=29, (* Electric potential (W/A) V *)
  Ohm            :=30, (* Electric resistance (VA) *)
  Joule          :=31, (* Energy (Nm) J *)
  Newtown        :=32, (* Force (kg m/s²) N *)
  Hertz          :=33, (* Frequency (1/s) Hz *)
  Lux            :=34, (* Illuminance (lm/m²) lx *)
  Lumen          :=35, (* Luminous flux (cd sr) Lm *)
  Weber          :=36, (* Magnetic flux (Vs) Wb *)
  Tesla          :=37, (* Magnetic flux density (Wb/m²) T *)
  Watt           :=38, (* Power (J/s) W *)
  Pascal         :=39, (* Pressure (N/m²) Pa *)

(* Extended units: *)
  SquareMeter    :=41, (* Area (m²) m² *)
  CubicMeter     :=42, (* Volume (m³) m³ *)
  MetersPerSecond :=43, (* Velocity (m/s) m/s *)
  MetersPerSecondSquare :=44, (* Acceleration (m/s²) m/s² *)
  CubicMetersPerSecond :=45, (* Volumetric flow rate (m³/s) m³/s *)
  MetersPerCubicMeter :=46, (* Fuel efficiency (m/m³) m/m³ *)
  KilogramMeter  :=47, (* Moment of mass (kg m) M *)
  KilogramPerCubicMeter :=48, (* Density (kg/m³) kg/m³ *)
  MeterSquarePerSecond :=49, (* Viscosity (m²/s) m²/s *)
```

```
WattPerMeterKelvin      :=50, (* Thermal conductivity (W/m K) W/m K *)
JoulePerKelvin          :=51, (* Heat capacity (J/K) J/K *)
PartsPerMillion         :=52, (* Concentration ppm *)
RotationsPerSecond     :=53, (* Rotational speed (1/s) 1/s *)
RadianPerSecond        :=54, (* Angular velocity (rad/s) rad/s *)

(* Industry specific units: *)
VoltAmpere              :=61, (* Apparent power (VA) VA *)
Watts                   :=62, (* Real power (I2R) W *)
VoltAmpereReactive      :=63, (* Reactive power (VISin) VAR *)
DegreesPh               :=64, (* Phase angle *)
DimensionlessPF         :=65, (* Power factor Cos *)
VoltSeconds             :=66, (* Volt seconds (Ws/A) Vs *)
VoltSquare              :=67, (* Volts squared (W2/A2) V2 *)
AmpSecond               :=68, (* Amp seconds (As) As *)
AmpSquare               :=69, (* Amps squared (A2) A2 *)
AmpSquareSecond        :=70, (* Amps squared time (A2s) A2t *)
VoltAmpereHours         :=71, (* Apparent energy VAh *)
WattHours               :=72, (* Real energy Wh *)
VoltAmpereReactiveHours :=73, (* Reactive energy VARh *)
VoltPerHertz            :=74  (* Magnetic flux V/Hz *)

) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|----------|-------------------------|
| None | Dimensionless |
| Meter | Length m |
| Kilogram | Mass kg |
| Second | Time s |
| Ampere | Current A |
| Kelvin | Temperature K |
| Mole | Amount of substance mol |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

6.6.75 E_AcsiSptEndSt

Namensraum: [Tc3_Acsi](#) [[▶](#) 96]
Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiSptEndSt :
(
    EndedNormally                := 1,
    EndedWithOvershoot           := 2,
    MeasurementWasDeviating      := 3,
    LossOfCommunicationWithDispatchCentre := 4,
    LossOfCommunicationWithLocalAreaNetwork := 5,
    LossOfCommunicationWithTheLocalInterface := 6,
    Timeout                      := 7,
    Voluntarily                  := 8,
    NoisyEnvironments            := 9,
    MaterialFailure              := 10,
    NewSetpointRequest           := 11,
    ImproperEnvironment          := 12,
    StabilityTimeWasReached       := 13,
    ImmobilisationTimeWasReached := 14,
    EquipmentInWrongMode         := 15,
    Unknown                      := 16
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|--|--------------|
| EndedNormally | |
| EndedWithOvershoot | |
| MeasurementWasDeviating | |
| LossOfCommunicationWithDispatchCentre | |
| LossOfCommunicationWithLocalAreaNetwork | |
| LossOfCommunicationWithTheLocalInterface | |
| Timeout | |
| Voluntarily | |
| NoisyEnvironments | |
| MaterialFailure | |
| NewSetpointRequest | |
| ImproperEnvironment | |
| StabilityTimeWasReached | |
| ImmobilisationTimeWasReached | |
| EquipmentInWrongMode | |
| Unknown | |

6.6.76 E_AcsiStClcTun

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiStClcTun :
(
    NotTuned                    := 1,
    Tuned                       := 2,
    TunedButNotCompensated      := 3,
    Umax                        := 4,
    UmaxButNotCompensated       := 5,
    UmaxButNotCompensatedDueToUContinuousLimitation := 6
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|---|--------------|
| NotTuned | |
| Tuned | |
| TunedButNotCompensated | |
| Umax | |
| UmaxButNotCompensated | |
| UmaxButNotCompensatedDueToUContinuousLimitation | |

6.6.77 E_AcsiStrWeekDay

Namensraum: [Tc3_Acsi \[► 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiStrWeekDay :
(
  Monday    := 1,
  Tuesday   := 2,
  Wednesday := 3,
  Thursday  := 4,
  Friday    := 5,
  Saturday  := 6,
  Sunday    := 7
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|-----------|--------------|
| Monday | Montag |
| Tuesday | Dienstag |
| Wednesday | Mittwoch |
| Thursday | Donnerstag |
| Friday | Freitag |
| Saturday | Samstag |
| Sunday | Sonntag |

6.6.78 E_AcsiSwOpCap

Namensraum: [Tc3_Acsi \[► 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiSwOpCap :
(
  None           := 1,
  Open           := 2,
  Close          := 3,
  OpenAndClose  := 4
) SINT;
END_TYPE
```


Werte

| Name | Beschreibung |
|--------------|--------------|
| None | |
| Open | |
| Close | |
| OpenAndClose | |

6.6.79 E_AcsiSwTyp

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiSwTyp :
(
  LoadBreakSwitch      := 1,
  Disconnecter         := 2,
  EarthingSwitch       := 3,
  HighSpeedEarthingSwitch := 4
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|-------------------------|--------------|
| LoadBreakSwitch | |
| Disconnecter | |
| EarthingSwitch | |
| HighSpeedEarthingSwitch | |

6.6.80 E_AcsiTcmd

Kontrollwert für einen Stufenschalter. Der Aufzählungswert wird in 2 Bits codiert. Die restlichen 6 Bits werden nicht verwendet.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiTcmd :
(
  Stop      := 2#00,
  Lower     := 2#01,
  Higher    := 2#10,
  {attribute 'hide'}reserved := 2#11
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|----------|----------------|
| Stop | Stoppen |
| Lower | Dekrementieren |
| Higher | Inkrementieren |
| reserved | Reserviert |

6.6.81 E_AcsiTmSyn

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiTmSyn :
(
    NotSynchronized           := 0,
    SynchronizedByLocalAreaClockSignal := 1,
    SynchronizedByGlobalAreaClockSignal := 2
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|-------------------------------------|--------------|
| NotSynchronized | |
| SynchronizedByLocalAreaClockSignal | |
| SynchronizedByGlobalAreaClockSignal | |

6.6.82 E_AcsiTnkTyp

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiTnkTyp :
(
    PressureOnly           := 1,
    LevelOnly              := 2,
    BothPressureAndLevel   := 3
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|----------------------|--------------|
| PressureOnly | |
| LevelOnly | |
| BothPressureAndLevel | |

6.6.83 E_AcsiTpcRxMod

Namensraum: [Tc3_Acsi](#) [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiTpcRxMod :
(
    Unused           := 1,
    Blocking         := 2,
    Permissive       := 3,
    Direct           := 4,
    Unblocking       := 5,
```

```
Status := 6
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|------------|--------------|
| Unused | |
| Blocking | |
| Permissive | |
| Direct | |
| Unblocking | |
| Status | |

6.6.84 E_AcsiTpcTxMod

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiTpcTxMod :
(
  Unused := 1,
  Blocking := 2,
  Permissive := 3,
  Direct := 4,
  Unblocking := 5,
  Status := 6
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|------------|--------------|
| Unused | |
| Blocking | |
| Permissive | |
| Direct | |
| Unblocking | |
| Status | |

6.6.85 E_AcsiTrBeh

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiTrBeh :
(
  SinglePoleTripping := 1,
  Undefined := 2,
  ThreePoleTripping := 3
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|--------------------|--------------|
| SinglePoleTripping | |
| Undefined | |
| ThreePoleTripping | |

6.6.86 E_AcsiTrgMod

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiTrgMod :
(
  Internal_ := 1,
  External_ := 2,
  Both      := 3
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|-----------|--------------|
| Internal_ | |
| External | |
| Both | |

6.6.87 E_AcsiTrMod

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiTrMod :
(
  ThreePhaseTripping      := 1,
  OneOrThreePhaseTripping := 2,
  Specific                 := 3,
  OnePhaseTripping        := 4
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|-------------------------|--------------|
| ThreePhaseTripping | |
| OneOrThreePhaseTripping | |
| Specific | |
| OnePhaseTripping | |

6.6.88 E_AcsiTypRsCrv

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiTypRsCrv :
(
    None := 1,
    DefiniteTimeDelayedReset := 2,
    InverseReset := 3
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|--------------------------|--------------|
| None | |
| DefiniteTimeDelayedReset | |
| InverseReset | |

6.6.89 E_AcsiUnbDetMth

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiUnbDetMth :
(
    NegativeSequence := 1,
    ZeroSequence := 2,
    NegativeSequencePositiveSequence := 3,
    ZeroSequencePositiveSequenceDirect := 4,
    PhaseVectorsComparison := 5,
    Others := 6
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|------------------------------------|--------------|
| NegativeSequence | |
| ZeroSequence | |
| NegativeSequencePositiveSequence | |
| ZeroSequencePositiveSequenceDirect | |
| PhaseVectorsComparison | |
| Others | |

6.6.90 E_AcsiUnBlkMod

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiUnBlkMod :
(
    Off := 1,
    Permanent := 2,
    TimeWindow := 3
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|------------|--------------|
| Off | |
| Permanent | |
| TimeWindow | |

6.6.91 E_AcsiVLanPriority

Virtual LAN tag control information: priority code point.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiVLanPriority :
(
  BK := 0,
  BE := 1,
  EE := 2,
  CA := 3,
  VI := 4,
  VO := 5,
  IC := 6,
  NC := 7
) BYTE;
END_TYPE
```

Werte

| Name | Beschreibung |
|------|-------------------------------|
| BK | Background (0). |
| BE | Best Effort (1). |
| EE | Excellent Effort (2). |
| CA | Critical Applications (3) |
| VI | Video (4), < 100 ms delay. |
| VO | Voice (5), < 10 ms delay. |
| IC | Internetnetwork Control (6). |
| NC | Network Control (7), highest. |

6.6.92 E_AcsiWeekDay

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiWeekDay :
(
  { attribute 'hide'}    reserved := 0,
  Monday    := 1,
  Tuesday   := 2,
  Wednesday := 3,
  Thursday  := 4,
  Friday    := 5,
  Saturday  := 6,
  Sunday    := 7
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|-----------|--------------|
| reserved | Reserviert |
| Monday | Montag |
| Tuesday | Dienstag |
| Wednesday | Mittwoch |
| Thursday | Donnerstag |
| Friday | Freitag |
| Saturday | Samstag |
| Sunday | Sonntag |

6.6.93 E_AcsiWeiMod

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AcsiWeiMod :
(
  Off           := 1,
  Operate       := 2,
  Echo          := 3,
  EchoAndOperate := 4
) SINT;
END_TYPE
```

Werte

| Name | Beschreibung |
|----------------|--------------|
| Off | |
| Operate | |
| Echo | |
| EchoAndOperate | |

6.6.94 E_AsyncEnvironmentState

Umgebungszustand der Kommunikationsverbindung.

Namensraum: [Tc3_Collections \[▶ 96\]](#)

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AsyncEnvironmentState :
(
  Idle           := 0,
  OutgoingConnect := 1,
  IncomingConnect := 2,
  Established     := 3,
  OutgoingDisconnect := 4,
  IncomingDisconnect := 5,
  Aborting       := 6
) BYTE;
END_TYPE
```

Werte

| Name | Beschreibung |
|--------------------|---|
| Idle | Ruhezustand. Verbindung ist geschlossen. |
| OutgoingConnect | Eine ausgehende (Client) Verbindung wird aufgebaut. |
| IncomingConnect | Eine eingehende (Server) Verbindung wird aufgebaut. |
| Established | Verbindung ist hergestellt. |
| OutgoingDisconnect | Eine (Client) Verbindung wird abgebaut. |
| IncomingDisconnect | Eine (Server) Verbindung wird abgebaut. |
| Aborting | Ein Abbruch der Verbindung wird durchgeführt. |

6.6.95 E_AsyncServiceState

Asynchroner Dienststatus.

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_AsyncServiceState :
(
    Completed := 0,
    Busy      := 1,
    Failed    := 2,
    Aborted   := 3
) BYTE;
END_TYPE
```

Werte

| Name | Beschreibung |
|-----------|--|
| Completed | Dienst wurde ausgeführt ohne Fehler. |
| Busy | Dienst wird gerade ausgeführt. |
| Failed | Dienst wurde beendet mit einem Fehler. |
| Aborted | Dienst wurde abgebrochen durch den Abort-Request-Befehl aus der Benutzerapplikation oder Abort-Indication aus der Kommunikationsschicht. |

6.6.96 E_EventCompletion

Ausführungstatus der Ereignisbehandlungsroutine.

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_EventCompletion :
(
    Completed := 0,
    Pending   := 1,
    Idle      := 2
) BYTE;
END_TYPE
```


Werte

| Name | Beschreibung |
|-----------|---|
| Completed | Ausgeführt (kompletiert). Auf die Bestätigung (Response) wird nicht gewartet. |
| Pending | Die Ausführung steht noch an und es wird auf die Bestätigung (Response) gewartet. |
| Idle | Die Ereignisbehandlungsroutine ist nicht implementiert. |

6.6.97 E_GseDispatchMode

GSE-Netzwerkadapter-Empfangsmodus. Dieser bestimmt, ob die empfangenden Frames auch an das Betriebssystem zur Weiterverarbeitung gesendet werden sollen.

Namensraum: [Tc3_Gse \[► 97\]](#)

Bibliothek: Tc3_Gse (Tc3_Gse.compiled-library)

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_GseDispatchMode :
(
    NonPromiscuous := 0,
    Monitor         := 1
) BYTE;
END_TYPE
```

Werte

| Name | Beschreibung |
|----------------|--|
| NonPromiscuous | Frames, die an den GSE-Netzwerkadapter adressiert und verarbeitet wurden (Broadcast- und Multicast-Frames inklusive), werden nicht mehr an das Betriebssystem zur Weiterverarbeitung gesendet. |
| Monitor | Alle empfangenen Frames werden an das Betriebssystem zur Weiterverarbeitung gesendet. |

6.6.98 E_GseDispatchMode

Namensraum: [Tc3_Gse \[► 97\]](#)

Bibliothek: Tc3_Gse (Tc3_Gse.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_GseDispatchMode :
(
    NonPromiscuous:= 0,
    Monitor         := 1
) BYTE;
END_TYPE
```

Werte

| Name | Beschreibung |
|----------------|--|
| NonPromiscuous | Empfangene Ethernet-Telegramme werden nicht an das Betriebssystem weitergegeben. |
| Monitor | Empfangene Ethernet-Telegramme werden an das Betriebssystem weitergegeben und können weiterverarbeitet werden. |

6.6.99 E_GseGoCBPublisherState

GOOSE-Steuerblock Publisher-Status.

Namensraum: [Tc3_Gse \[► 97\]](#)

Bibliothek: Tc3_Gse (Tc3_Gse.compiled-library)

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_GseGoCBPublisherState :
(
  NonExistent      := 0,
  SendValues       := 1,
  RetransmitPending := 2,
  Retransmit       := 3
) BYTE;
END_TYPE
```

Werte

| Name | Beschreibung |
|-------------------|--|
| NonExistent | Deaktiviert. |
| SendValues | Aktiv und sendet GOOSE-Nachricht (Update inklusive). |
| RetransmitPending | Aktiv und wartet auf GOOSE-Nachrichtwiederholung. |
| Retransmit | Aktiv und sendet GOOSE-Nachrichtwiederholung. |

6.6.100 E_GseGoCBSubscriberState

GOOSE-Steuerblock Subscriber-Status.

Namensraum: [Tc3_Gse \[► 97\]](#)

Bibliothek: Tc3_Gse (Tc3_Gse.compiled-library)

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_GseGoCBSubscriberState :
(
  Off           := 0,
  NonExistent  := 1,
  Valid        := 2,
  Questionable := 3
) BYTE;
END_TYPE
```

Werte

| Name | Beschreibung |
|--------------|--|
| Off | Deaktiviert. |
| NonExistent | Aktiv und hat noch keine GOOSE-Nachrichten empfangen. |
| Valid | Aktiv und empfängt kontinuierlich GOOSE-Nachrichten. |
| Questionable | Aktiv und die GOOSE-Nachrichten werden gar nicht oder zu spät empfangen (erlaubte Lebenszeit/Gültigkeitsdauer ist abgelaufen). |

6.6.101 E_GseLinkStatus

GSE-Netzwerkadapter Link-Status.

Namensraum: [Tc3_Gse \[► 97\]](#)

Bibliothek: Tc3_Gse (Tc3_Gse.compiled-library)

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_GseLinkStatus :
(
  Ok      := 0,
```

```

LinkError := 1
);
END_TYPE

```

Werte

| Name | Beschreibung |
|-----------|----------------------------|
| Ok | Der Link ist hergestellt. |
| LinkError | Der Link ist unterbrochen. |

6.6.102 E_GseRetransmissionStrategy

Zeitalgorithmus für die wiederholten GOOSE-Meldungen.

Namensraum: [Tc3_Gse \[► 97\]](#)

Bibliothek: Tc3_Gse (Tc3_Gse.compiled-library)

```

{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_GseRetransmissionStrategy :
(
    Arithmetic:=0
) BYTE;
END_TYPE

```

Werte

| Name | Beschreibung |
|------------|---|
| Arithmetic | Die erste Meldung wird nach der minimalen Intervallzeit gesendet. Jede weitere Meldung (Wiederholung) wird nach dem Zeitintervall gesendet, der ein Produkt der vorherigen Intervallzeit und des Multiplikators ist, bis zum Erreichen der Maximalzeit. Bei einer Wertänderung wird eine GOOSE-Meldung sofort gesendet und die Intervallzeit erneut auf den minimalen Wert zurückgesetzt. |

6.6.103 E_MmsVmdLogicalStatus

MMS Typ: vmdLogicalStatus

Namensraum: [Tc3_Mms \[► 97\]](#)

Bibliothek: Tc3_Mms (Tc3_Mms.compiled-library)

Syntax

```

{attribute 'qualified_only'}
TYPE E_MmsVmdLogicalStatus :
(
    StateChangesAllowed      :=0,
    NoStateChangesAllowed    :=1,
    LimitedServicesPermitted :=2,
    SupportServicesAllowed   :=3
) BYTE;
END_TYPE

```

6.6.104 E_MmsVmdPhysicalStatus

MMS Typ: vmdPhysicalStatus.

Namensraum: [Tc3_Mms \[► 97\]](#)

Bibliothek: Tc3_Mms (Tc3_Mms.compiled-library)

Syntax

```

{attribute 'qualified_only'}
TYPE E_MmsVmdPhysicalStatus :
(
    Operational      :=0,

```

```

    PartiallyOperational :=1,
    Inoperable           :=2,
    NeedsCommissioning   :=3
) BYTE;
END_TYPE

```

6.6.105 E_Rfc1006TpduSize

Maximale TPDU Bytelänge.

Namensraum: [Tc3_Rfc1006](#) [[▶ 97](#)]

Bibliothek: Tc3_Rfc1006 (Tc3_Rfc1006.compiled-library)

Syntax

```

TYPE E_Rfc1006TpduSize :
(
    Size_0      := 0,
    Size_128    := 7,
    Size_256    := 8,
    Size_512    := 9,
    Size_1024   := 10,
    Size_2048   := 11,
    Size_4096   := 12,
    Size_8192   := 13,
    Size_16384  := 14,
    Size_32768  := 15
) BYTE;
END_TYPE

```

Werte

| Name | Beschreibung |
|------------|---------------------|
| Size_0 | Nicht spezifiziert. |
| Size_128 | 128 Byte (default). |
| Size_256 | 256 Byte. |
| Size_512 | 512 Byte. |
| Size_1024 | 1024 Byte. |
| Size_2048 | 2048 Byte. |
| Size_4096 | 4096 Byte. |
| Size_8192 | 8192 Byte. |
| Size_16384 | 16384 Byte. |
| Size_32768 | 32768 Byte. |

6.6.106 E_ScsmEdition

Namensraum: [Tc3_iec61850_8_1](#) [[▶ 97](#)]

Bibliothek: Tc3_iec61850_8_1 (Tc3_iec61850_8_1.compiled-library)

Syntax

```

{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_ScsmEdition :
(
    Ed1 := 1,
    Ed2 := 2
) BYTE := Ed2;
END_TYPE

```

Werte

| Name | Beschreibung |
|------|-----------------------|
| Ed1 | DIN EN 61850-8-1:2005 |
| Ed2 | DIN EN 61850-8-1:2011 |

6.6.107 E_TraceLevel

Prioritätsstufe für Protokollierung der Fehlermeldungen und Logmeldungen.

Namensraum: [Tc3_Collections](#) [► 96]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_TraceLevel :
(
  None      := 0, // don't trace any message
  Verbose   := 16#00000001, // bit 0 => log verbose messages
  TxData    := 16#00000002, // bit 1 => log tx data
  RxData    := 16#00000004, // bit 2 => log rx data
  Info      := 16#00000100, // bit 8..15 => trace pdu info messages
  PduReq    := 16#00000200,
  PduInd    := 16#00000400,
  PduRsp    := 16#00000800,
  PduCnf    := 16#00001000,
  Warning   := 16#00010000, // bit 16..23 => trace warning messages
  Error     := 16#01000000, // bit 24..30 => trace error messages
  Critical  := 16#80000000 // bit 31 => trace critical messages
) DWORD;
END_TYPE
```

Werte

| Name | Beschreibung |
|----------|------------------|
| None | Keine |
| Verbose | Text |
| TxData | Gesendete Daten |
| RxData | Empfangene Daten |
| Info | Information |
| PduReq | PDU-Request |
| PduInd | PDU-Indication |
| PduRsp | PDU-Response |
| PduCnf | PDU-Confirmation |
| Warning | Warnung |
| Error | Fehler |
| Critical | Kritisch |

6.6.108 E_UtcTimeAccuracy

Anzahl der signifikanten Bits in der Membervariablen: [T_UtcTime](#) [► 479].fractionOfSecond. Werte 25 bis 30 werden nicht verwendet.

Namensraum: [Tc3_Collections](#) [► 96]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_UtcTimeAccuracy :
```

```
(  
  Null      := 0,  
  _01      := 1,  
  _02      := 2,  
  _03      := 3,  
  _04      := 4,  
  _05      := 5,  
  _06      := 6,  
  _07      := 7,  
  _08      := 8,  
  _09      := 9,  
  _10      := 10,  
  _11      := 11,  
  _12      := 12,  
  _13      := 13,  
  _14      := 14,  
  _15      := 15,  
  _16      := 16,  
  _17      := 17,  
  _18      := 18,  
  _19      := 19,  
  _20      := 20,  
  _21      := 21,  
  _22      := 22,  
  _23      := 23,  
  _24      := 24,  
  _25_invalid := 2#11001,  
  _26_invalid := 2#11010,  
  _27_invalid := 2#11011,  
  _28_invalid := 2#11100,  
  _29_invalid := 2#11101,  
  _30_invalid := 2#11110,  
  Unspecified := 31  
) BYTE;  
END_TYPE
```

Werte

| Name | Beschreibung |
|--------------------------|--------------------|
| Null | 0 Bits |
| _01 | 1 Bit |
| _02 | 2 Bit |
| _03 | 3 Bit |
| _04 | 4 Bit |
| _05 | 5 Bit |
| _06 | 6 Bit |
| _07 | 7 Bit |
| _08 | 8 Bit |
| _09 | 9 Bit |
| _10 | 10 Bit |
| _11 | 11 Bit |
| _12 | 12 Bit |
| _13 | 13 Bit |
| _14 | 14 Bit |
| _15 | 15 Bit |
| _16 | 16 Bit |
| _17 | 17 Bit |
| _18 | 18 Bit |
| _19 | 19 Bit |
| _20 | 20 Bit |
| _21 | 21 Bit |
| _22 | 22 Bit |
| _23 | 23 Bit |
| _24 | 24 Bit |
| _25_invalid.._30_invalid | Ungültig |
| Unspecified | Nicht spezifiziert |

6.6.109 Externe Datentypen

6.6.109.1 ETHERNET_ADDRESS

Netzwerkadapter MAC-Adresse.

Namensraum: [TwinCAT TypeSystem \[► 96\]](#)

Bibliothek: Nicht erforderlich. TwinCAT Systemresource ab v3.1.4024.0

Syntax

```
TYPE ETHERNET_ADDRESS :
  STRUCT
    b : ARRAY [0..5] OF BYTE;
  END_STRUCT
END_TYPE
```

6.6.109.2 GUID

System ID.

Namensraum: [TwinCAT TypeSystem \[► 96\]](#)

Bibliothek: Nicht erforderlich. TwinCAT Systemresource ab v3.1.4024.0

Syntax

```

TYPE GUID :
STRUCT
  Data1 : DWORD;
  Data2 : WORD;
  Data3 : WORD;
  Data4 : ARRAY[0..7] OF BYTE;
END_STRUCT
END_TYPE

```

6.6.109.3 OTCID

Objekt-ID des TwinCAT Realtime Netzwerkadapters.

Namensraum: [TwinCAT TypeSystem](#) [► 96]

Bibliothek: Nicht erforderlich. TwinCAT Systemresouce ab v3.1.4024.0

```

TYPE OTCID : UDINT;
END_TYPE

```

6.6.109.4 T_AmsNetID

Eine SPS-Variable von diesem Typ ist ein String, der die AMS-Netzwerkennung des Zielgerätes enthält, an das der ADS-Befehl gerichtet wird. Der String besteht aus sechs, durch Punkte getrennten Zahlenfeldern. Gültige AMS-Netzwerkadressen sind z. B. '1.1.1.2.7.1' oder '200.5.7.170.1.7'. Wird ein Leerstring übergeben, so wird automatisch die AMS-Netzwerkennung des lokalen Gerätes angenommen.

Namensraum: Tc2_System

Bibliothek: Tc2_System (Tc2_System.compiled-library)

```

TYPE T_AmsNetID : STRING(23);
END_TYPE

```

6.6.109.5 T_HSOCKET

TCP/IP-Socket-Verbindungshandle.

Namensraum: Tc2_Tcplp

Bibliothek: Tc2_Tcplp (Tc2_Tcplp.compiled-library)

Syntax

```

TYPE T_HSOCKET

```

6.6.109.6 T_IPv4Addr

Eine Variable von diesem Typ ist ein String mit der (Ipv4) Internet-Protocol-Netzwerkadresse. Z. B. '172.16.7.199'.

```

TYPE T_IPv4Addr : STRING(15);
END_TYPE

```

Voraussetzungen

| Entwicklungsumgebung | Zielplattform | Einzubindende SPS-Bibliotheken (Kategoriegruppe) |
|----------------------|----------------------------|--|
| TwinCAT v3.1.0 | PC oder CX (x86, x64, ARM) | Tc2_System (System) |

6.6.109.7 T_MaxString

Eine Variable von diesem Typ ist ein String mit maximaler Länge. Längere Strings dürfen benutzt werden, die String-Funktionen können aber zur Zeit nur 255 Zeichen bearbeiten.

```

TYPE T_MaxString : STRING(MAX_STRING_LENGTH);
END_TYPE

```



```
VAR_GLOBAL CONSTANT
    MAX_STRING_LENGTH : UDINT := 255;
END_VAR
```

Voraussetzungen

| Entwicklungsumgebung | Zielplattform | Einzubindende SPS-Bibliotheken (Kategoriegruppe) |
|----------------------|----------------------------|--|
| TwinCAT v3.1.0 | PC oder CX (x86, x64, ARM) | Tc2_System (System) |

6.6.109.8 T_ThrottleTimes

Zykluszeiten für den Empfangsdaten-Throttle-Polling-Mode.

Namensraum: Tc2_Tcplp

Bibliothek: Tc2_Tcplp (Tc2_Tcplp.compiled-library)

Syntax

```
TYPE T_ThrottleTimes
```

6.6.109.9 TcEthernetAdapterPriority

Priorität des TwinCAT Realtime-Netzwerkadapters.

Namensraum: [TwinCAT TypeSystem \[► 96\]](#)

Bibliothek: Nicht erforderlich. TwinCAT Systemresouce ab v3.1.4024.0

```
TYPE TcEthernetAdapterPriority :
(
    TcEthernetAdapterPriorityLow :=1,
    TcEthernetAdapterPriorityHigh :=16
);
END_TYPE
```

6.6.109.10 TcEventEntry

Strukturierter Typ mit Ereignis- oder Fehlerinformation.

Namensraum: [TwinCAT TypeSystem \[► 96\]](#)

Bibliothek: Nicht erforderlich. TwinCAT Systemresouce ab v3.1.4024.0

Syntax

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

Komponenten

| Name | Typ | Beschreibung |
|----------------|---|---|
| uuidEventClass | GUID [► 451] | Eine eindeutige Kennung der Ereignisklasse. Die GUID [► 451] spezifiziert die Ereignisquelle. |
| nEventID | UDINT | Ereignis-ID oder Fehlernummer. |
| eSeverity | TcEventSeverity [► 453] | Schweregrad. |

6.6.109.11 TcEventSeverity

Schweregrad einer Meldung.

Namensraum: [TwinCAT TypeSystem \[► 96\]](#)

Bibliothek: Nicht erforderlich. TwinCAT Systemresource ab v3.1.4024.0

Syntax

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

Werte

| Name | Beschreibung |
|----------|--------------|
| Verbose | Text |
| Info | Information |
| Warning | Warnung |
| Error | Fehler |
| Critical | Kritisch |

6.6.109.12 TIMESTRUCT

Uhrzeit im Systemzeitformat.

```
TYPE TIMESTRUCT
STRUCT
  wYear      : WORD;
  wMonth     : WORD;
  wDayOfWeek : WORD;
  wDay       : WORD;
  wHour      : WORD;
  wMinute    : WORD;
  wSecond    : WORD;
  wMilliseconds : WORD;
END_STRUCT
END_TYPE
```

wYear: Das Jahr: 1970 ~ 2106;

wMonth: Der Monat: 1 ~ 12 (Januar = 1, Februar = 2 usw.);

wDayOfWeek: Der Wochentag: 0 ~ 6 (Sonntag = 0, Montag = 1 usw.);

wDay: Tag des Monats: 1 ~ 31;

wHour: Stunde: 0 ~ 23;

wMinute: Minute: 0 ~ 59;

wSecond: Sekunde: 0 ~ 59;

wMilliseconds: Millisekunde: 0 ~ 999;

Voraussetzungen

| Entwicklungsumgebung | Zielplattform | Einzubindende SPS-Bibliotheken (Kategoriegruppe) |
|----------------------|----------------------------|--|
| TwinCAT v3.1.0 | PC oder CX (x86, x64, ARM) | Tc2_Uilities (System) |

6.6.110 ST_AcsiAnalogueValue

Analoger Wert als strukturierter Typ.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
TYPE ST_AcsiAnalogueValue :
STRUCT
    i : DINT;
    f : REAL;
END_STRUCT
END_TYPE
```

Komponenten

| Name | Typ | Beschreibung |
|------|------|--------------------|
| i | DINT | Ganzzahliger Wert. |
| f | REAL | Fließkommawert. |

6.6.111 ST_AcsiCalendarTime

Kalenderzeit als strukturierter Typ.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
TYPE ST_AcsiCalendarTime :
STRUCT
    occ : WORD;
    occType : E_AcsiOccType;
    occPer : E_AcsiOccPer;
    weekDay : E_AcsiWeekDay;
    month : E_AcsiMonth;
    day : BYTE;
    hr : BYTE;
    mn : BYTE;
END_STRUCT
END_TYPE
```

Komponenten

| Name | Typ | Beschreibung |
|---------|---------------------------------------|-----------------------------------|
| occ | WORD | Auftreten eines Kalenderelements. |
| occType | E_AcsiOccType [▶ 417] | Art des Kalenderelements. |
| occPer | E_AcsiOccPer [▶ 417] | Wiederholungszeitraum. |
| weekDay | E_AcsiWeekDay [▶ 442] | Wochentag. |
| month | E_AcsiMonth [▶ 415] | Monat. |
| day | BYTE | Tag. |
| hr | BYTE | Stunde. |
| mn | BYTE | Minute. |

6.6.112 ST_AcsiCell

Zelle als strukturierter Typ.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

TYPE ST_AcsiCell :
STRUCT
  xStart : REAL;
  xEnd   : REAL;
  yStart : REAL;
  yEnd   : REAL;
END_STRUCT
END_TYPE

```

Komponenten

| Name | Typ | Beschreibung |
|--------|------|-------------------------|
| xStart | REAL | Zellkoordinate x-Start. |
| xEnd | REAL | Zellkoordinate x-Ende. |
| yStart | REAL | Zellkoordinate y-Start. |
| yEnd | REAL | Zellkoordinate y-Ende. |

6.6.113 ST_AcsiDataSetClassInfo

Datensatz-Member Container-Information.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

```

TYPE ST_AcsiDataSetClassInfo :
STRUCT
  nMembers      : UDINT;
  bPersistent   : BOOL;
  bDeletable    : BOOL;
END_STRUCT
END_TYPE

```

Komponenten

| Name | Typ | Beschreibung |
|-------------|-------|--|
| nMembers | UDINT | Anzahl der Datensatz-Member. |
| bPersistent | BOOL | TRUE => Persistenter Datensatz (gebunden an einen LN). FALSE => Nicht-Persistenter Datensatz (gebunden an aktive Verbindung, wird gelöscht beim Verbindungsabbruch). |
| bDeletable | BOOL | TRUE => Datensatz wurde dynamisch konfiguriert (zur Laufzeit und kann gelöscht werden). FALSE => Datensatz wurde statisch konfiguriert (kann nicht gelöscht werden). |

6.6.114 ST_AcsiFunctionalConstraints

Funktionale Gruppe als strukturierter Typ.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

TYPE ST_AcsiFunctionalConstraints :
STRUCT
  MX : BIT;
  ST : BIT;
  CO : BIT;
  CF : BIT;
  DC : BIT;
  SP : BIT;
  SG : BIT;
  RP : BIT;
  LG : BIT;
  BR : BIT;
  GO : BIT;

```

```

GS : BIT;
SV : BIT;
SE : BIT;
MS : BIT;
SC : BIT;
US : BIT;
EX : BIT;
// XX : BIT;
END_STRUCT
END_TYPE

```

Komponenten

| Name | Typ | Beschreibung |
|------|-----|--|
| MX | BIT | Measurands (analogue values) (use of FC in the definition of DATA). |
| ST_ | BIT | Status information (use of FC in the definition of DATA). |
| CO | BIT | Control (use of FC in the definition of DATA). |
| CF | BIT | Configuration (use of FC in the definition of DATA). |
| DC | BIT | Description (use of FC in the definition of DATA). |
| SP | BIT | Setpoint (use of FC in the definition of DATA and control blocks). |
| SG | BIT | Setting group (use of FC in the definition of DATA). |
| RP | BIT | Unbuffered report (-
> Reserved for control classes and use of the FC in the definition of control blocks). |
| LG | BIT | Logging . |
| BR | BIT | Buffered report (-
> Reserved for control classes and use of the FC in the definition of control blocks). |
| GO | BIT | Goose control (-
> Reserved for control classes and use of the FC in the definition of control blocks). |
| GS | BIT | Gsse control (-
> Reserved FOR control classes AND use OF the FC in the definition OF control blocks). |
| SV | BIT | Substitution (use of FC in the definition of DATA). |
| SE | BIT | Setting group editable (use of FC in the definition of DATA). |
| MS | BIT | Multicast sampled value control (-
> Reserved for control classes and use of the FC in the definition of control blocks). |
| SC | BIT | SCL. |
| US | BIT | Unicast sampled value control (-
> Reserved for control classes and use of the FC in the definition of control blocks). |
| EX | BIT | Extended definition (use of FC in the definition of DATA). |
| XX | BIT | Shall represent all DataAttributes of a Data of any FC. This value shall only used in the FCD; "XX" shall not used as value in DataAttributes. |

6.6.115 ST_AcsiIEDGroupClassInfo

Container-Information der IED-Gruppe.

Namensraum: Tc3_Acsi [▶ 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

```

TYPE ST_AcsiIEDGroupClassInfo :
STRUCT
    nIEDs : UDINT;
END_STRUCT
END_TYPE

```

Komponenten

| Name | Typ | Beschreibung |
|-------|-------|-------------------------------|
| nIEDs | UDINT | Anzahl der IEDs im Container. |

6.6.116 ST_AcsiIntelligentElectronicDeviceClassInfo

Container-Information des IEDs.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

```
TYPE ST_AcsiIntelligentElectronicDeviceClassInfo :
STRUCT
    nLogicalDevices : UDINT;
    nAaNodes        : UDINT;
END_STRUCT
END_TYPE
```

Komponenten

| Name | Typ | Beschreibung |
|-----------------|-------|--|
| nLogicalDevices | UDINT | Anzahl der Logischen Geräte im Container des Servers. |
| nAaNodes | UDINT | Anzahl der assoziierten Verbindungen im Container des Servers. |

6.6.117 ST_AcsiLogicalDeviceClassInfo

Container-Information des Logischen-Gerätes.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

```
TYPE ST_AcsiLogicalDeviceClassInfo :
STRUCT
    nLogicalNodes : UDINT;
END_STRUCT
END_TYPE
```

Komponenten

| Name | Typ | Beschreibung |
|---------------|-------|---|
| nLogicalNodes | UDINT | Anzahl der Logischen Knoten im Container des Logischen-Gerätes. |

6.6.118 ST_AcsiLogicalNodeClassInfo

Container-Information des Logischen-Knoten.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

```
TYPE ST_AcsiLogicalNodeClassInfo :
STRUCT
    nDdatas          : UDINT;
    nDataSets       : UDINT;
    nBufferedReportControlBlocks : UDINT;
    nUnbufferedReportControlBlocks : UDINT;
    nLogControlBlocks : UDINT;
    nSettingGroupControlBlocks : UDINT;
    nGooseControlBlocks : UDINT;
    nGsseControlBlocks : UDINT;
    nMulticastSampledValueControlBlocks : UDINT;
    nUnicastSampledValueControlBlocks : UDINT;
END_STRUCT
END_TYPE
```

Komponenten

| Name | Typ | Beschreibung |
|-------------------------------------|-------|--|
| nDats | UDINT | Anzahl der Daten und Datenattribute im Container des Logischen-Knoten. |
| nDataSets | UDINT | Anzahl der Data-Sets im Container des Logischen-Knoten. |
| nBufferedReportControlBlocks | UDINT | Anzahl der BRCBs im Container des Logischen-Knoten. |
| nUnbufferedReportControlBlocks | UDINT | Anzahl der URCBs im Container des Logischen-Knoten. |
| nLogControlBlocks | UDINT | Anzahl der LCBs im Container des Logischen-Knoten. |
| nSettingGroupControlBlocks | UDINT | Anzahl der SGCBs im Container des Logischen-Knoten. |
| nGooseControlBlocks | UDINT | Anzahl der GOCBs im Container des Logischen-Knoten. |
| nGsseControlBlocks | UDINT | Anzahl der GSCBs im Container des Logischen-Knoten. |
| nMulticastSampledValueControlBlocks | UDINT | Anzahl der MSVCBs im Container des Logischen-Knoten. |
| nUnicastSampledValueControlBlocks | UDINT | Anzahl der USVCBs im Container des Logischen-Knoten. |

6.6.119 ST_AcsiLogOptionalFields

Diese Struktur informiert über die optionalen Felder, die in einem Log übertragen werden.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
TYPE ST_AcsiLogOptionalFields :
STRUCT
    ReasonForInclusion : BIT;
END_STRUCT
END_TYPE
```

Komponenten

| Name | Typ | Beschreibung |
|--------------------|-----|---|
| ReasonForInclusion | BIT | Bei TRUE wird ein Log mit der Ursache für die Berichterstattung übertragen. |

6.6.120 ST_AcsiOptionalFields

Diese Struktur informiert über die optionalen Felder die in einem Report übertragen werden.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
TYPE ST_AcsiOptionalFields :
STRUCT
{ attribute 'hide' } reserved : BIT;
    SequenceNumber : BIT;
    ReportTimeStamp : BIT;
    ReasonForInclusion : BIT;
    DataSetName : BIT;
    DataReference : BIT;
    BufferOverflow : BIT;
    EntryID : BIT;
    ConfRevision : BIT;
    Segmentation : BIT;
END_STRUCT
END_TYPE
```

Komponenten

| Name | Typ | Beschreibung |
|--------------------|-----|--|
| reserved | BIT | Reserviert. |
| SequenceNumber | BIT | Bei TRUE wird der Report mit einer Sequenznummer übertragen. |
| ReportTimeStamp | BIT | Bei TRUE wird der Report mit einem Zeitstempel übertragen. |
| ReasonForInclusion | BIT | Bei TRUE wird der Report mit der Ursache für die Berichterstattung übertragen. |
| DataSetName | BIT | Bei TRUE wird in dem Report auch der Name des jeweils referenzierten Datensatzes übertragen. |
| DataReference | BIT | Bei TRUE werden auch die Objektreferenzen der Datensatzmember im Report übertragen. |
| BufferOverflow | BIT | Bei TRUE wird das Attribut für den Pufferüberlauf im Report übertragen. |
| EntryID | BIT | Bei TRUE erhalten die Einträge im Report eine fortlaufende Nummer. |
| ConfRevision | BIT | Bei TRUE wird das Attribut für die Revision der Konfiguration im Report übertragen. |
| Segmentation | BIT | Dieses Bit ist reserviert. |

6.6.121 ST_AcsiOriginator

Information über den Befehlsauftraggeber/Urheber als strukturierter Typ.

Namensraum: [Tc3_Acsi \[► 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

TYPE ST_AcsiOriginator :
STRUCT
    orCat    : E_AcsiOrCategory;
    orIdent  : T_OCTET64;
END_STRUCT
END_TYPE

```

Komponenten

| Name | Typ | Beschreibung |
|---------|--|---|
| orCat | E_AcsiOrCategory [► 419] | Kategorie des Befehlsauftraggebers/Urhebers. |
| orIdent | T_OCTET64 [► 479] | Identifikation des Befehlsauftraggebers/Urhebers. |

6.6.122 ST_AcsiPhyComAddr

Diese Struktur beschreibt eine physikalische Kommunikationsadresse.

Namensraum: [Tc3_Acsi \[► 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

TYPE ST_AcsiPhyComAddr :
STRUCT
    Addr      : T_OCTET6;
    PRIORITY  : BYTE;
    VID       : WORD;
    APPID     : WORD;
END_STRUCT
END_TYPE

```


Komponenten

| Name | Typ | Beschreibung |
|----------|----------|-------------------------------------|
| Addr | T_OCTET6 | Media Access Control (MAC) address. |
| PRIORITY | BYTE | Anwenderpriorität. |
| VID | WORD | VLAN-Identifikation. |
| APPID | WORD | Anwendungsidentifikation. |

6.6.123 ST_AcsiPoint

Punkt als strukturierter Typ.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

TYPE ST_AcsiPoint :
STRUCT
  xVal : REAL;
  yVal : REAL;
  zVal : REAL;
END_STRUCT
END_TYPE
    
```

Komponenten

| Name | Typ | Beschreibung |
|------|------|--------------------|
| xVal | REAL | Punktcoordinate x. |
| yVal | REAL | Punktcoordinate y. |
| zVal | REAL | Punktcoordinate z. |

6.6.124 ST_AcsiPresCond

Diese Struktur konfiguriert Systemweit die Verfügbarkeit der optionalen Datenattribute.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

TYPE ST_AcsiPresCond :
STRUCT
  AnalogueValue           : E_AcsiAnalogueValuePresCond;
  Cell_xEnd                : BIT := 0;
  Cell_yStart              : BIT := 0;
  Cell_yEnd                : BIT := 0;
  Point_zVal               : BIT := 0;
  RangeConfig_limDb       : BIT := 0;
  Unit_multiplier          : BIT := 0;
  ValWithTrans_transInd   : BIT := 0;
  Vector_ang               : BIT := 0;
  Oper_AnalogueValue_operTm : BIT := 0;
  Oper_BOOLEAN_operTm     : BIT := 0;
  Oper_CODEENUM_operTm    : BIT := 0;
  Oper_ENUMERATED_operTm  : BIT := 0;
  Oper_INT32_operTm        : BIT := 0;
  Oper_INT8_operTm        : BIT := 0;
  SBOw_AnalogueValue_operTm : BIT := 0;
  SBOw_BOOLEAN_operTm     : BIT := 0;
  SBOw_CODEENUM_operTm    : BIT := 0;
  SBOw_ENUMERATED_operTm  : BIT := 0;
  SBOw_INT32_operTm        : BIT := 0;
  SBOw_INT8_operTm        : BIT := 0;
  Cancel_AnalogueValue_operTm : BIT := 0;
  Cancel_BOOLEAN_operTm   : BIT := 0;
  Cancel_CODEENUM_operTm  : BIT := 0;
    
```

```

Cancel_ENUMERATED_operTm : BIT := 0;
Cancel_INT32_operTm      : BIT := 0;
Cancel_INT8_operTm       : BIT := 0;
BRCB_ResvTms             : BIT := 0;
BRCB_Owner                : BIT := 0;
URCB_Owner               : BIT := 0;
LCB_OptFlds              : BIT := 0;
LCB_BufTm                 : BIT := 0;
END_STRUCT
END_TYPE

```

Komponenten

| Name | Typ | Beschreibung |
|-----------------------------|---|---|
| AnalogueValue | E_AcsiAnalogueValuePresCond [▶ 397] | Konfiguriert die Präsenz der AnalogueValue „i“ und „f“ attribute in den Kontrollwerten (ctlVal) und Statuswerten (mxVal). |
| Cell_xEnd | BIT | False 0 := optional (default);
True 1 := mandatory; |
| Cell_yStart | BIT | -//- |
| Cell_yEnd | BIT | -//- |
| Point_zVal | BIT | -//- |
| RangeConfig_limDb | BIT | -//- |
| Unit_multiplier | BIT | -//- |
| ValWithTrans_transInd | BIT | -//- |
| Vector_ang | BIT | -//- |
| Oper_AnalogueValue_operTm | BIT | -//- |
| Oper_BOOLEAN_operTm | BIT | -//- |
| Oper_CODEDENUM_operTm | BIT | -//- |
| Oper_ENUMERATED_operTm | BIT | -//- |
| Oper_INT32_operTm | BIT | -//- |
| Oper_Int8_operTm | BIT | -//- |
| SBOw_AnalogueValue_operTm | BIT | -//- |
| SBOw_BOOLEAN_operTm | BIT | -//- |
| SBOw_CODEDENUM_operTm | BIT | -//- |
| SBOw_ENUMERATED_operTm | BIT | -//- |
| SBOw_INT32_operTm | BIT | -//- |
| SBOw_INT8_operTm | BIT | -//- |
| Cancel_AnalogueValue_operTm | BIT | -//- |
| Cancel_BOOLEAN_operTm | BIT | -//- |
| Cancel_CODEDENUM_operTm | BIT | -//- |
| Cancel_ENUMERATE_operTm | BIT | -//- |
| Cancel_INT32_operTm | BIT | -//- |
| Cancel_INT8_operTm | BIT | -//- |
| BRCB_ResvTms | BIT | -//- |
| BRCB_Owner | BIT | -//- |
| URCB_Owner | BIT | -//- |
| LCB_OptFlds | BIT | -//- |
| LCB_BufTm | BIT | -//- |

6.6.125 ST_AcsiPulseConfig

Pulse-Konfiguration als strukturierter Typ.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

TYPE ST_AcsiPulseConfig :
STRUCT
    cmdQual : E_AcsiCmdQual;
    onDur   : DWORD;
    offDur  : DWORD;
    numPls  : DWORD;
END_STRUCT
END_TYPE
    
```

Komponenten

| Name | Typ | Beschreibung |
|---------|---------------------------------------|-------------------|
| cmdQual | E_AcsiCmdQual [▶ 404] | Befehlskriterium. |
| onDur | DWORD | Pulsdauer: ON. |
| offDur | DWORD | Pulsdauer: OFF |
| numPls | DWORD | Anzahl der Pulse. |

6.6.126 ST_AcsiQuality

Quality als strukturierter Typ.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

TYPE ST_AcsiQuality :
STRUCT
    Validity0      : BIT;
    Validity1      : BIT;
    Overflow        : BIT;
    OutOfRange     : BIT;
    BadReference    : BIT;
    Oscillatory     : BIT;
    Failure         : BIT;
    OldData         : BIT;
    Inconsistent    : BIT;
    Inaccurate      : BIT;
    Source          : BIT;
    Test           : BIT;
    OperatorBlocked : BIT;
END_STRUCT
END_TYPE
    
```

Komponenten

| Name | Typ | Beschreibung |
|-----------------|-----|--|
| Validity0 | BIT | Gültigkeit LSB. Wertekodierung: siehe Tabelle unten. |
| Validity1 | BIT | Gültigkeit MSB. Wertekodierung: siehe Tabelle unten. |
| Overflow | BIT | Überlauf. |
| OutOfRange | BIT | Nicht im zulässigen Bereich. |
| BadReference | BIT | Falscher Bezug. |
| Oscillatory | BIT | Oszillierend. |
| Failure | BIT | Misserfolg. |
| OldData | BIT | Alte Daten. |
| Inconsistent | BIT | Inkonsistent. |
| Inaccurate | BIT | Ungenau. |
| Source | BIT | Quelle (0:=Prozess, 1:=Ersetzt). |
| Test | BIT | Test. |
| OperatorBlocked | BIT | Bediener blockiert. |

Validity-Wertekodierung

| Wert | Validity0 | Validity1 | Beschreibung |
|--------------|-----------|-----------|---------------------|
| Good | 0 | 0 | Gut (Standardwert). |
| Invalid | 0 | 1 | Ungültig. |
| Reserved | 1 | 0 | Reserviert. |
| Questionable | 1 | 1 | Fraglich. |

6.6.127 ST_AcsiRangeConfig

Bereichskonfiguration als strukturierter Typ.

Namensraum: [Tc3_Acsi](#) [► 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

TYPE ST_AcsiRangeConfig :
STRUCT
  hhLim : ST_AcsiAnalogueValue;
  hLim  : ST_AcsiAnalogueValue;
  lLim  : ST_AcsiAnalogueValue;
  llLim : ST_AcsiAnalogueValue;
  min_  : ST_AcsiAnalogueValue;
  max_  : ST_AcsiAnalogueValue;
  limDb : DWORD;
END_STRUCT
END_TYPE

```

Komponenten

| Name | Typ | Beschreibung |
|-------|--|------------------|
| hhLim | ST_AcsiAnalogueValue [► 455] | High-High-Limit. |
| hLim | ST_AcsiAnalogueValue [► 455] | High-Limit. |
| lLim | ST_AcsiAnalogueValue [► 455] | Low-Limit. |
| llLim | ST_AcsiAnalogueValue [► 455] | Low-Low-Limit. |
| min_ | ST_AcsiAnalogueValue [► 455] | Minimalwert. |
| max_ | ST_AcsiAnalogueValue [► 455] | Maximalwert. |
| limDb | DWORD | |

6.6.128 ST_AcsiReasonCode

Übertragungsursache eines Listenelements in einem Report. Ein Report kann einen, mehrere oder alle Listenelemente eines Datensatzes beinhalten. Jedes Listenelement kann eine eigene Übertragungsursache besitzen.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

TYPE ST_AcsiReasonCode :
STRUCT
{ attribute 'hide'}reserved : BIT;
  DataChange : BIT;
  QualityChange : BIT;
  DataUpdate : BIT;
  Integrity : BIT;
  GeneralInterrogation : BIT;
  ApplicationTrigger : BIT;
END_STRUCT
END_TYPE
    
```

Komponenten

| Name | Typ | Beschreibung |
|----------------------|-----|--|
| reserved | BIT | Reserviert. |
| DataChange | BIT | Wenn TRUE, dann wurde der Report wegen einer Datenänderung gesendet. |
| QualityChange | BIT | Wenn TRUE, dann wurde der Report wegen einer Qualitätsänderung gesendet. |
| DataUpdate | BIT | Wenn TRUE, dann wurde der Report wegen einer Datenaktualisierung gesendet. Im Unterschied zu der Option: „DataChange“ wird der Report auch dann gesendet wenn die Daten gleich geblieben sind, aber erneut gesetzt wurden. |
| Integrity | BIT | Wenn TRUE, dann wurde der Report wegen dem Ablauf der Integritätsdauer gesendet. Die Datensätze werden in diesem Fall regelmäßig nach Ablauf der Integritätsdauer gesendet. |
| GeneralInterrogation | BIT | Wenn TRUE, dann wurde der Report wegen einer einmaligen Generalabfrage der Datensätze gesendet. |
| ApplicationTrigger | BIT | Wenn TRUE, dann wurde der Report durch eine Funktion in der Applikation ausgelöst/gesendet. |

6.6.129 ST_AcsiScaledValueConfig

Skalierter-Wert-Konfiguration als strukturierter Typ.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

TYPE ST_AcsiScaledValueConfig :
STRUCT
  scaleFactor : REAL;
  offset : REAL;
END_STRUCT
END_TYPE
    
```

Komponenten

| Name | Typ | Beschreibung |
|-------------|------|--------------------|
| scaleFactor | REAL | Skalierungsfaktor. |
| offset | REAL | Offset. |

6.6.130 ST_AcsiSvOptionalFields

Diese Struktur informiert über die optionalen Felder, die mit einem abgetasteten Wert übertragen werden.

Namensraum: [Tc3_Acsi](#) [► 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

TYPE ST_AcsiSvOptionalFields :
STRUCT
    RefreshTime          : BIT;
    SampleSynchronized  : BIT;
    SampleRate           : BIT;
    DataSetName          : BIT;
    Security              : BIT;
    SampleMode           : BIT;
    SynchSourceIdentity  : BIT;
END_STRUCT
END_TYPE

```

Komponenten

| Name | Typ | Beschreibung |
|---------------------|-----|---|
| RefreshTime | BIT | Beim TRUE wird der abgetastete Wert mit dem Attributwert „RefrTm“ übertragen. |
| SampleSynchronized | BIT | Dieser Wert wird ignoriert und existiert wegen der Kompatibilität zu IEC 61850-9-2 Edition 1.0. |
| SampleRate | BIT | Beim TRUE wird der abgetastete Wert mit dem Attributwert „SmpRate“ übertragen. |
| DataSetName | BIT | Beim TRUE wird der abgetastete Wert mit dem Attributwert „DatSet“ übertragen. |
| Security | BIT | Beim TRUE wird der abgetastete Wert mit dem Attributwert „Security“ übertragen. |
| SampleMode | BIT | Beim TRUE wird der abgetastete Wert mit dem Attributwert „SmpMod“ übertragen. |
| SynchSourceIdentity | BIT | Beim TRUE wird der abgetastete Wert mit dem Attributwert „GmlIdentity“ übertragen. |

6.6.131 ST_AcsiTriggerConditions

Auslösoptionen für das Senden eines Reports. Mehrere Optionen dürfen gleichzeitig gesetzt werden. Die Optionen „Integrity“ und „GeneralInterrogation“ dürfen nicht gleichzeitig gesetzt werden.

Namensraum: [Tc3_Acsi](#) [► 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

TYPE ST_AcsiTriggerConditions :
STRUCT
{ attribute 'hide'}reserved : BIT;
    DataChange              : BIT;
    QualityChange           : BIT;
    DataUpdate              : BIT;
    Integrity                : BIT;
    GeneralInterrogation    : BIT;

```

```
{ attribute 'hide'}reserved2 : BIT;
END_STRUCT
END_TYPE
```

Komponenten

| Name | Typ | Beschreibung |
|----------------------|-----|--|
| reserved | BIT | Reserviert. |
| DataChange | BIT | Beim TRUE wird das Senden eines Reports wegen einer Datenänderung aktiviert. |
| QualityChange | BIT | Beim TRUE wird das Senden eines Reports wegen einer Qualitätsänderung aktiviert. |
| DataUpdate | BIT | Beim TRUE wird das Senden eines Reports wegen einer Datenaktualisierung aktiviert. Im Unterschied zu der Option „DataChange“ wird der Report auch dann gesendet, wenn die Daten gleich geblieben sind, aber erneut gesetzt wurden. |
| Integrity | BIT | Beim TRUE wird das Senden eines Reports wegen dem Ablauf der Integritätsdauer aktiviert. Die Datensätze werden in diesem Fall regelmäßig nach Ablauf der Integritätsdauer gesendet. |
| GeneralInterrogation | BIT | Beim TRUE wird das Senden eines Reports wegen einer einmaligen Generalabfrage der Datensätze aktiviert. |
| reserved2 | BIT | Reserviert, um das Alignment mit „ReasonCode“ abzugleichen. |

6.6.132 ST_AcsiUnit

Messwerteinheit als strukturierter Typ.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
TYPE ST_AcsiUnit :
STRUCT
    SIUnit      : E_AcsiSIUnit;
    multiplier  : E_AcsiMultiplier;
END_STRUCT
END_TYPE
```

Komponenten

| Name | Typ | Beschreibung |
|------------|--|----------------|
| SIUnit | E_AcsiSIUnit [▶ 432] | SI-Einheit. |
| multiplier | E_AcsiMultiplier [▶ 416] | Multiplikator. |

6.6.133 ST_AcsiValWithTrans

Wert mit Übergangszustand als strukturierter Typ.

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
TYPE ST_AcsiValWithTrans :
STRUCT
    posVal     : SINT;
```

```

    transInd : BOOL;
END_STRUCT
END_TYPE

```

Komponenten

| Name | Typ | Beschreibung |
|----------|------|-------------------|
| posVal | SINT | Position. |
| transInd | BOOL | Übergangszustand. |

6.6.134 ST_AcsiVector

Vektor als strukturierter Typ.

Namensraum: [Tc3_Acsi \[► 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```

TYPE ST_AcsiVector :
STRUCT
    mag : ST_AcsiAnalogueValue;
    ang : ST_AcsiAnalogueValue;
END_STRUCT
END_TYPE

```

Komponenten

| Name | Typ | Beschreibung |
|------|--|--------------|
| mag | ST_AcsiAnalogueValue [► 455] | Größe. |
| ang | ST_AcsiAnalogueValue [► 455] | Winkel. |

6.6.135 ST_GseAdapterInfo

Status und Statistikdaten des GSE-Netzwerkadapters.

Namensraum: [Tc3_Gse \[► 97\]](#)

Bibliothek: Tc3_Gse (Tc3_Gse.compiled-library)

```

TYPE ST_GseAdapterInfo :
STRUCT
    eStatus : E_GseLinkStatus;
    nRx : UDINT;
    nTx : UDINT;
END_STRUCT
END_TYPE

```

Komponenten

| Name | Typ | Beschreibung |
|---------|---|----------------------------------|
| eStatus | E_GseLinkStatus [► 446] | GSE-Netzwerkadapter Link-Status. |
| nRx | UDINT | Anzahl der empfangenen Frames. |
| nTx | UDINT | Anzahl der gesendeten Frames. |

6.6.136 ST_GseGoCBPublisherTx

Information in der zuletzt gesendeten GOOSE-Nachricht.

Namensraum: [Tc3_Gse \[► 97\]](#)

Bibliothek: Tc3_Gse (Tc3_Gse.compiled-library)


```

TYPE ST_GseGoCBPublisherTx :
STRUCT
  eState          : E_GseGoCBPublisherState;
  (*-----*)
  srcAddr         : ETHERNET_ADDRESS;
  dstAddr         : ETHERNET_ADDRESS;
  nAPPID          : WORD;
  nVID            : WORD(0..4095);
  ePRIORITY       : E_AcsiVlanPriority;
  (*-----*)
  sGoCBRef        : T_AcsiObjectReference;
  nTimeAllowedToLive : UDINT;
  sDatSet         : T_AcsiObjectReference;
  sGoID           : T_AcsiVisString129;
  tT              : T_UtcTime;
  nStNum          : UDINT;
  nSqNum          : UDINT;
  bTest           : BOOL;
  nConfRev        : UDINT;
  bNdsCom         : BOOL;
  nNumDatSetEntries : UDINT;
  (*-----*)
  nMinTime        : UDINT(1..16#7FFFFFFF);
  nMaxTime        : UDINT(1..16#7FFFFFFF);
  nMultiplier     : BYTE(1..16#FF);
  nTimeRemainingToLive : UDINT;

  ipAdapter       : I_GseAdapterClass;
END_STRUCT
END_TYPE
    
```

Komponenten

| Name | Typ | Beschreibung |
|----------------------|--|---|
| eState | E_GseGoCBPublisherState
[▶ 446] | Publisher-Status. |
| srcAddr | ETHERNET_ADDRESS [▶ 451] | MAC-Adresse der Nachrichtenquelle. |
| dstAddr | ETHERNET_ADDRESS [▶ 451] | MAC-Adresse des Nachrichtenziels. |
| nAPPID | WORD | APPID. |
| nVID | WORD(0..4095) | Virtual LAN Identifikation. |
| ePRIORITY | E_AcsiVlanPriority [▶ 442] | Virtual LAN Priorität. |
| sGoCBRef | T_AcsiObjectReference [▶ 476] | GOOSE-Steuerblock Referenzstring. |
| nTimeAllowedToLive | UDINT | Erlaubte Lebenszeit/Gültigkeitsdauer der GOOSE-Nachricht [ms]. |
| sDatSet | T_AcsiObjectReference [▶ 476] | Data-Set-Referenzstring. |
| sGoID | T_AcsiVisString129 | GOOSE-ID. |
| tT | T_UtcTime [▶ 479] | Zeitstempel (UTC). |
| nStNum | UDINT | Statusnummer. |
| nSqNum | UDINT | Sequenznummer. |
| bTest | BOOL | Simulation-/Test-Flag. |
| nConfRev | UDINT | Konfiguration- Revisionsnummer. |
| bNdsCom | BOOL | Konfiguration/Parametrierung benötigt. |
| nNumDatSetEntries | UDINT | Anzahl der Data-Set-Einträge. |
| nMinTime | UDINT(1..16#7FFFFFFF) | Anfangszeit für Nachrichtenwiederholungen [ms]. |
| nMaxTime | UDINT(1..16#7FFFFFFF) | Finale Zeit für Nachrichtenwiederholungen [ms]. |
| nMultiplier | BYTE(1..16#FF) | Multiplikator für Nachrichtenwiederholungen [ms]. |
| nTimeRemainingToLive | UDINT | Verbliebene Lebenszeit/Gültigkeitsdauer der GOOSE-Nachricht [ms]. |
| ipAdapter | I_GseAdapterClass | GSE-Netzwerkadapter-Schnittstellenzeiger. |

6.6.137 ST_GseGoCBSubscriberRx

Information in der zuletzt empfangenen GOOSE-Nachricht.

Namensraum: Tc3_Gse [[▶ 97](#)]

Bibliothek: Tc3_Gse (Tc3_Gse.compiled-library)

```

TYPE ST_GseGoCBSubscriberRx :
STRUCT
  eState          : E_GseGoCBSubscriberState;
  (*-----*)
  srcAddr        : ETHERNET_ADDRESS;
  dstAddr        : ETHERNET_ADDRESS;
  nAPPID         : WORD;
  nVID           : WORD(0..4095);
  ePRIORITY      : E_AcsiVlanPriority;
  (*-----*)
  sGoCBRef       : T_AcsiObjectReference;
  nTimeAllowedToLive : UDINT;
  sDatSet        : T_AcsiObjectReference;
  sGoID          : T_AcsiVisString129;
  tT             : T_UtcTime;
  nStNum         : UDINT;
  nSqNum         : UDINT;
  bTest          : BOOL;
  nConfRev       : UDINT;
  bNdsCom        : BOOL;
  nNumDatSetEntries : UDINT;
  (*-----*)
  nTimeRemainingToLive : UDINT;
  ipAdapter      : I_GseAdapterClass;
END_STRUCT
END_TYPE

```

Komponenten

| Name | Typ | Beschreibung |
|----------------------|---|---|
| eState | E_GseGoCBSubscriberState
▶ 446 | Subscriber-Status. |
| srcAddr | ETHERNET_ADDRESS ▶ 451 | MAC-Adresse der Nachrichtenquelle. |
| dstAddr | ETHERNET_ADDRESS ▶ 451 | MAC-Adresse des Nachrichtenziels. |
| nAPPID | WORD | APPID. |
| nVID | WORD | Virtual LAN Identifikation. |
| ePRIORITY | E_AcsiVlanPriority ▶ 442 | Virtual LAN Priorität. |
| sGoCBRef | T_AcsiObjectReference ▶ 476 | GOOSE-Steuerblock Referenzstring. |
| nTimeAllowedToLive | UDINT | Erlaubte Lebenszeit/Gültigkeitsdauer der GOOSE-Nachricht [ms]. |
| sDatSet | T_AcsiObjectReference ▶ 476 | Data-Set-Referenzstring. |
| sGoID | T_AcsiVisString129 | GOOSE-ID. |
| tT | T_UtcTime ▶ 479 | Zeitstempel (UTC). |
| nStNum | UDINT | Statusnummer. |
| nSqNum | UDINT | Sequenznummer. |
| bTest | BOOL | Simulation-/Test-Flag. |
| nConfRev | UDINT | Konfiguration- Revisionsnummer. |
| bNdsCom | BOOL | Konfiguration/Parametrierung benötigt. |
| nNumDatSetEntries | UDINT | Anzahl der Data-Set-Einträge. |
| nTimeRemainingToLive | UDINT | Verbliebene Lebenszeit/Gültigkeitsdauer der GOOSE-Nachricht [ms]. |
| ipAdapter | I_GseAdapterClass | GSE-Netzwerkadapter-Schnittstellenzeiger. |

6.6.138 ST_ScsmAdditionalCauseDiagnosticEventInfo

Diese Struktur liefert zusätzliche Diagnoseinformationen zu einer negativen Rückmeldung während der Ausführung eines Dienstes für die Steuerung (Operate, Select, Cancel, usw.). Für die Übertragung dieser Information ist bei der Verwendung des MMS Mappings der Dienst MMS-InformationReport vorgesehen.

Namensraum: [Tc3_iec61850_8_1](#) [[▶ 97](#)]

Bibliothek: Tc3_iec61850_8_1 (Tc3_iec61850_8_1.compiled-library)

Syntax

```

TYPE ST_ScsmAdditionalCauseDiagnosticEventInfo :
STRUCT
    eService : E_AcsiServiceType;
    CntrlObj : T_AcsiObjectReference;
    Error : E_AcsiApplError;
    Origin : ST_AcsiOriginator;
    CtlNum : USINT;
    AddCause : E_AcsiApplAddCause;
END_STRUCT
END_TYPE
    
```

Komponenten

| Name | Typ | Beschreibung |
|----------|---|---|
| eService | E_AcsiServiceType [▶ 429] | Dienstbezeichnung, dessen Ausführung die zusätzliche Diagnoseinformation generiert hat. |
| CntrlObj | T_AcsiObjectReference [▶ 476] | String mit der Objektreferenz des Objekts (Dienstes für die Steuerung), der bei der Ausführung die zusätzliche Diagnoseinformation generiert hat. Bei einem Leerstring ist dieser Parameter optional und nicht verfügbar. |
| Error | E_AcsiApplError [▶ 399] | Diagnoseinformation über die Fehlerquelle (Zustandsmaschine des Dienstes, der Operate- oder TimeActivated-Dienst selbst oder andere). |
| Origin | ST_AcsiOriginator [▶ 460] | Bezeichnet den Client, der die Ausführung des Dienstes veranlasst hat. |
| CtlNum | USINT | Diese Nummer entspricht der Dienst-Sequenznummer des Clients, der die Ausführung des Dienstes veranlasst hat. Beim Wert 0 ist dieser Parameter optional und nicht verfügbar. |
| AddCause | E_AcsiApplAddCause [▶ 398] | Zusätzliche Information über die Ursache. |

6.6.139 ST_ScsmBrCBJobInfo

BRCB Statistik- und Status-Information.

Namensraum: [Tc3_iec61850_8_1](#) [[▶ 97](#)]

Bibliothek: Tc3_iec61850_8_1 (Tc3_iec61850_8_1.compiled-library)

```

TYPE ST_ScsmBrCBJobInfo :
STRUCT
    eState : E_ScsmRCBState; (* Client status. *)
    sRptID : T_AcsiVisString129;
    sDatSet : T_AcsiObjectReference;
    nSqNum : WORD;
    tTimeOfEntry : T_BinaryTime;
    stReasonCode : ST_AcsiReasonCode;
    nEntryID : LWORD;
    ipAA : I_ScsmAssociationClass;
END_STRUCT
END_TYPE
    
```

6.6.140 ST_ScsmBrCBValuesSet

Auswahl der BRCB-Steuerblock Attribute die geschrieben werden sollen.

Syntax

```

TYPE ST_ScsmBrCBValuesSet :
STRUCT
  RptID      : BIT; (* R/W *)
  RptEna     : BIT; (* R/W *)
  DatSet     : BIT; (* R/W *)
  ConfRev    : BIT; (* Read only attribute! *)
  OptFlds    : BIT; (* R/W *)
  BufTm      : BIT; (* R/W *)
  SqNum      : BIT; (* Read only attribute! *)
  TrgOps     : BIT; (* R/W *)
  IntgPd     : BIT; (* R/W *)
  GI         : BIT; (* R/W *)
  PurgeBuf   : BIT; (* R/W *)
  EntryID    : BIT; (* R/W *)
  TimeOfEntry : BIT; (* Read only attribute! *)
  ResvTms    : BIT; (* R/W *)
  Owner      : BIT; (* Read only attribute! *)
END_STRUCT
END_TYPE

```

6.6.141 ST_ScsmCommandTerminationEventInfo

Diese Struktur liefert zusätzliche Informationen zur Dienstrückmeldung: CommandTermination.

Namensraum: [Tc3_iec61850_8_1](#) [[▶ 97](#)]

Bibliothek: [Tc3_iec61850_8_1](#) (Tc3_iec61850_8_1.compiled-library)

Syntax

```

TYPE ST_ScsmCommandTerminationEventInfo :
STRUCT
  eService : E_AcsiServiceType;
  CntrlObj  : T_AcsiObjectReference;
  OperTm   : T_UtcTime;
  Origin    : ST_AcsiOriginator;
  CtlNum    : USINT;
  T         : T_UtcTime;
  Test     : BOOL;
  Check    : E_AcsiCheck;
END_STRUCT
END_TYPE

```

Komponenten

| Name | Typ | Beschreibung |
|----------|--|---|
| eService | E_AcsiServiceType
[▶ 429] | Dienstbezeichnung dessen Ausführung die Dienstrückmeldung generiert hat. |
| CntrlObj | T_AcsiObjectReference
[▶ 476] | String mit der Objektreferenz des Objekts (Dienstes für die Steuerung), der bei der Ausführung die zusätzliche Information generiert hat. Bei einem Leerstring ist dieser Parameter optional und nicht verfügbar. |
| OperTm | T_UtcTime [▶ 479] | |
| Origin | ST_AcsiOriginator
[▶ 460] | Bezeichnet den Client der die Ausführung des Dienstes veranlasst hat. |
| CtlNum | USINT | Diese Nummer entspricht der Dienst-Sequenznummer des Clients, der die Ausführung des Dienstes veranlasst hat. |
| T | T_UtcTime [▶ 479] | |
| Test | BOOL | |
| Check | E_AcsiCheck [▶ 401] | |

6.6.142 ST_ScsmConfirmedEventInfo

Diese Struktur liefert zusätzliche Informationen zur Dienstrückmeldung.

Namensraum: [Tc3_iec61850_8_1](#) [[▶ 97](#)]

Bibliothek: Tc3_iec61850_8_1 (Tc3_iec61850_8_1.compiled-library)

Syntax

```

TYPE ST_ScsmConfirmedEventInfo :
STRUCT
    eService      : E_AcsiServiceType;
    hUser         : PVOID;
    nInvokeID     : UDINT;
    nEventSequence : UDINT;
    bMoreFollows  : BOOL;
END_STRUCT
END_TYPE
    
```

Komponenten

| Name | Typ | Beschreibung |
|----------------|---|--|
| eService | E_AcsiServiceType [▶ 429] | Dienstbezeichnung dessen Ausführung die Dienstrückmeldung generiert hat. |
| hUser | PVOID | Optionaler benutzerdefinierter Parameter oder auch Zeiger. Ein beliebiger Wert für diesen Parameter kann in der Methoden für Dienstaktivierung (XyzMethodReq) übergeben werden. In der Ereignisbehandlungsroutine der Dienstrückmeldung wird dann der gleiche Wert gespiegelt. |
| nInvokeID | UDINT | Identifiziert die Dienstaktivierung. Jede neue Dienstaktivierung (XyzMethodReq) wird fortlaufend nummeriert. Eine Kopie dieser Nummer wird in der Dienstrückmeldung gespiegelt (OnXyzMethodCnf). |
| nEventSequence | UDINT | Sequenznummer der Ereignisbehandlungsroutine. Bei einem mehrfachen Aufruf der Ereignisbehandlungsroutine werden die Aufrufe fortlaufend nummeriert. |
| bMoreFollows | BOOL | FALSE => Letzter Aufruf der Ereignisbehandlungsroutine. TRUE => Weitere Aufrufe der Ereignisbehandlungsroutine werden folgen. |

6.6.143 ST_ScsmGoCBJobInfo

GOCB Statistik- und Status-Information.

Namensraum: [Tc3_iec61850_8_1](#) [[▶ 97](#)]

Bibliothek: Tc3_iec61850_8_1 (Tc3_iec61850_8_1.compiled-library)

```

TYPE ST_ScsmGoCBJobInfo :
STRUCT
    bEnabled : BOOL;
END_STRUCT
END_TYPE
    
```

6.6.144 ST_ScsmGoCBValuesSet

Auswahl der GOOSE-Steuerblock Attribute die geschrieben werden sollen.

Syntax

```

TYPE ST_ScsmGoCBValuesSet :
STRUCT
    GoEna      : BIT; (* R/W *)
    GoID       : BIT; (* R/W *)
    DatSet     : BIT; (* R/W *)
    ConfRev    : BIT; (* Read only attribute! *)
    NdsCom     : BIT;
    DstAddress : BIT;
    MinTime    : BIT;
    MaxTime    : BIT;
END_STRUCT
    
```

```

FixedOffs : BIT;
END_STRUCT
END_TYPE

```

6.6.145 ST_ScsmReportEventInfo

Diese Struktur liefert zusätzliche Informationen zum empfangenen Report.

Namensraum: Tc3_iec61850_8_1 [► 97]

Bibliothek: Tc3_iec61850_8_1 (Tc3_iec61850_8_1.compiled-library)

Syntax

```

TYPE ST_ScsmReportEventInfo :
STRUCT
    eService          : E_AcsiServiceType;
    RptID             : T_AcsiObjectReference;
    OptFlds           : ST_AcsiOptionalFields;
    SeqNum            : UINT;
    TimeOfEntry       : T_BinaryTime;
    DatSet            : T_AcsiObjectReference;
    BufOvfl           : BOOL;
    EntryID           : T_OCTET8;
    ConfRev           : UDINT;
    SubSeqNum         : UINT;
    MoreSegmentsFollow : BOOL;
    DataReference     : T_AcsiObjectReference;
    ReasonCode        : ST_AcsiReasonCode;
    nEventSequence    : UDINT;
    bMoreFollows      : BOOL;
END_STRUCT
END_TYPE

```

Komponenten

| Name | Typ | Beschreibung |
|--------------------|---|---|
| eService | E_AcsiServiceType [► 429] | Dienstbezeichnung dessen Ausführung den Report generiert hat. |
| RptID | T_AcsiObjectReference [► 476] | Eindeutige Reportkennung. |
| OptFlds | ST_AcsiOptionalFields [► 459] | Information über die optionalen Felder. |
| SeqNum | UINT | Report-Sequenznummer. |
| TimeOfEntry | T_BinaryTime [► 476] | Report-Zeitstempel. |
| DatSet | T_AcsiObjectReference [► 476] | Dataset-Referenz |
| BufOvfl | BOOL | Pufferüberlauf. |
| EntryID | T_OCTET8 [► 478] | Identifikation des Report-Eintrags. |
| ConfRev | UDINT | Revision der Konfiguration. |
| SubSeqNum | UINT | Subsequenznummer. |
| MoreSegmentsFollow | BOOL | Wenn TRUE, dann folgen weitere Segmente. |
| DataReference | T_AcsiObjectReference [► 476] | Datareferenz. |
| ReasonCode | ST_AcsiReasonCode [► 465] | Übertragungsursache. |
| nEventSequence | UDINT | Sequenznummer der Ereignisbehandlungsroutine. Bei einem mehrfachen Aufruf der Ereignisbehandlungsroutine werden die Aufrufe fortlaufend nummeriert. |
| bMoreFollows | BOOL | FALSE => Letzter Aufruf der Ereignisbehandlungsroutine. TRUE => Weitere Aufrufe der Ereignisbehandlungsroutine werden folgen. |

6.6.146 ST_ScsmUrCBJobInfo

URCB Statistik- und Status-Information.

Namensraum: [Tc3_iec61850_8_1](#) [[▶ 97](#)]

Bibliothek: Tc3_iec61850_8_1 (Tc3_iec61850_8_1.compiled-library)

```

TYPE ST_ScsmUrCBJobInfo :
STRUCT
  eState      : E_ScsmRCBState; (* Client status. *)
  sRptID     : T_AcsiVisString129;
  sDatSet    : T_AcsiObjectReference;
  nSqNum     : BYTE;
  tTimeOfEntry : T_BinaryTime;
  stReasonCode : ST_AcsiReasonCode;
  ipAA      : I_ScsmAssociationClass;
END_STRUCT
END_TYPE
    
```

6.6.147 ST_ScsmUrCBValuesSet

Auswahl der URCB-Steuerblock Attribute die geschrieben werden sollen.

Syntax

```

TYPE ST_ScsmUrCBValuesSet :
STRUCT
  RptID      : BIT; (* R/W *)
  RptEna     : BIT; (* R/W *)
  Resv       : BIT; (* R/W *)
  DatSet     : BIT; (* R/W *)
  ConfRev    : BIT; (* Read only attribute! *)
  OptFlds   : BIT; (* R/W *)
  BufTm     : BIT; (* R/W *)
  SqNum      : BIT; (* Read only attribute! *)
  TrgOps     : BIT; (* R/W *)
  IntgPd     : BIT; (* R/W *)
  GI         : BIT; (* R/W *)
  Owner      : BIT; (* Read only attribute! *)
END_STRUCT
END_TYPE
    
```

6.6.148 ST_ServiceErrorClass

Strukturierter Typ mit Ereignis- oder Fehlerinformation. Siehe: [TcEventEntry](#) [[▶ 453](#)].

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```

ALIAS ST_ServiceErrorClass : TcEventEntry;
    
```

Komponenten

| Name | Typ | Beschreibung |
|----------------|--|---|
| uuidEventClass | GUID | Ereignisklasse. Die GUID spezifiziert die Ereignisquelle. |
| nEventID | UDINT | Ereignis-ID oder Fehlernummer. |
| eSeverity | TcEventSeverity
[▶ 453] | Schweregrad |

6.6.149 T_AcsiObjectName

Objektname als String mit einer max. Länge von 64 Zeichen.

Namensraum: [Tc3_Acsi](#) [► 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
ALIAS T_AcsiObjectName : STRING(MAX_ACSI_NAME)
```

6.6.150 T_AcsiObjectReference

Objektreferenz als String mit einer max. Länge von 255 Zeichen.

Namensraum: [Tc3_Acsi](#) [► 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
ALIAS T_AcsiObjectReference : T_MaxString
```

6.6.151 T_AcsiTag

In einer Variablen von diesem Typ kann eine zusätzliche Information gespeichert werden (standardmäßig eine vorzeichenlose Zahl aber auch Pointer).

Namensraum: [Tc3_Acsi](#) [► 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
ALIAS T_AcsiTag : UXINT
```

6.6.152 T_AcsiVisString129

String mit einer max. Länge von 129 Zeichen.

Namensraum: [Tc3_Acsi](#) [► 96]

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
ALIAS T_AcsiVisString129 : STRING(129)
```

6.6.153 T_BinaryTime

Binary-Time Zeitinformation.

Namensraum: [Tc3_Collections](#) [► 96]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
TYPE T_BinaryTime :  
STRUCT  
    timeOfDay : TOD;  
    day       : WORD;  
{ attribute 'hide'}reserved : WORD := 0;  
END_STRUCT  
END_TYPE
```


Komponenten

| Name | Typ | Beschreibung |
|-----------|------|--|
| timeOfDay | TOD | Anzahl der vergangenen Millisekunden seit Mitternacht des aktuellen Tages. |
| day | WORD | Anzahl der vergangenen Tage seit dem 1. Januar 1984. |
| reserved | WORD | Reserviert (0). |

6.6.154 T_INT24

Vorzeichenbehaftete 24-Bit-Zahl.

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
ALIAS T_INT24 : ARRAY[0..2] OF BYTE
```

6.6.155 T_INT128

Vorzeichenbehaftete 128-Bit-Zahl.

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
TYPE T_INT128 :
STRUCT
  Lo : LWORD;
  Hi : LWORD;
END_STRUCT
END_TYPE
```

Komponenten

| Name | Typ | Beschreibung |
|------|-------|----------------|
| Lo | LWORD | Unteren Bytes. |
| Hi | LWORD | Oberen Bytes. |

6.6.156 T_MmsInteger8

Vorzeichenbehaftete 8 Bit Zahl.

Namensraum: [Tc3_Mms](#) [[▶ 97](#)]

Bibliothek: Tc3_Mms (Tc3_Mms.compiled-library)

Syntax

```
ALIAS T_MmsInteger8 : SINT(-128..127)
```

6.6.157 T_MmsInteger16

Vorzeichenbehaftete 16 Bit Zahl.

Namensraum: [Tc3_Mms](#) [[▶ 97](#)]

Bibliothek: Tc3_Mms (Tc3_Mms.compiled-library)

Syntax

```
ALIAS T_MmsInteger16 : INT(-32768..32767)
```

6.6.158 T_MmsInteger32

Vorzeichenbehaftete 32 Bit Zahl.

Namensraum: [Tc3_Mms](#) [[▶ 97](#)]

Bibliothek: Tc3_Mms (Tc3_Mms.compiled-library)

Syntax

```
ALIAS T_MmsInteger32 : DINT(-2147483648..2147483647)
```

6.6.159 T_MmsString

MMS Typ: MMSString (UTF8-String).

Namensraum: [Tc3_Mms](#) [[▶ 97](#)]

Bibliothek: Tc3_Mms (Tc3_Mms.compiled-library)

Syntax

```
ALIAS T_MmsString : T_MaxWString
```

6.6.160 T_MmsVisibleString

MMS Typ: VisibleString.

Namensraum: [Tc3_Mms](#) [[▶ 97](#)]

Bibliothek: Tc3_Mms (Tc3_Mms.compiled-library)

Syntax

```
ALIAS T_MmsVisibleString : T_MaxString
```

6.6.161 T_OCTET6

Oktetstring mit einer max. Länge von 6 Oktets.

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
ALIAS T_OCTET6 : ARRAY[0..5] OF BYTE
```

6.6.162 T_OCTET8

Oktetstring mit einer max. Länge von 8 Oktets.

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
ALIAS T_OCTET8 : ARRAY [0..7] OF BYTE
```

6.6.163 T_OCTET16

Oktetstring mit einer max. Länge von 16 Oktets.

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
ALIAS T_OCTET16 : ARRAY[0..15] OF BYTE
```

6.6.164 T_OCTET32

Oktetstring mit einer max. Länge von 32 Oktets.

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
ALIAS T_OCTET32 : ARRAY[0..31] OF BYTE
```

6.6.165 T_OCTET64

Oktetstring mit einer max. Länge von 64 Oktets.

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
ALIAS T_OCTET64 : ARRAY[0..63] OF BYTE
```

6.6.166 T_UINT24

Vorzeichenlose 24-Bit-Zahl.

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
ALIAS T_UINT24 : ARRAY[0..2] OF BYTE
```

6.6.167 T_UINT128

Vorzeichenlose 128-Bit-Zahl.

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```
TYPE T_UINT128 :
STRUCT
  Lo : LWORD;
  Hi : LWORD;
END_STRUCT
END_TYPE
```

Komponenten

| Name | Typ | Beschreibung |
|------|-------|---------------|
| Lo | LWORD | Untere Bytes. |
| Hi | LWORD | Obere Bytes. |

6.6.168 T_UtcTime

UTC-Time-Zeitinformation.

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```

TYPE T_UtcTime :
STRUCT
    secondSinceEpoch : DT;
    fractionOfSecond : T_UINT24;
    quality            : T_UtcTimeQuality := ( Accuracy0 := 0, Accuracy1 := 0, Accuracy2 := 0, Accurac
y3 := 0, Accuracy4 := 0 );
END_STRUCT
END_TYPE

```

Komponenten

| Name | Typ | Beschreibung |
|------------------|---|--|
| secondSinceEpoch | DT | Anzahl der vergangenen, ganzen Sekunden seit dem 1 Januar 1970. |
| fractionOfSecond | T_UINT24 [▶ 479] | Sekundenbruchteil codiert als 24-Bit-Zahl. Codierung:
Bit 0 := 0.5s
Bit 1 := 0.25s
Bit 2 := 0.125s
usw.
Die max. Auflösung beträgt ~60 Mikrosekunden. |
| quality | T_UtcTimeQuality
[▶ 480] | Zusätzliche Information über die Qualität und die Genauigkeit der Zeitinformation. |

6.6.169 T_UtcTimeQuality

Information über die Qualität und die Genauigkeit der UTC-Time-Zeitinformation.

Namensraum: [Tc3_Collections](#) [[▶ 96](#)]

Bibliothek: Tc3_Collections (Tc3_Collections.compiled-library)

Syntax

```

TYPE T_UtcTimeQuality :
STRUCT
    LeapSecondsKnown      : BIT;
    ClockFailure          : BIT;
    ClockNotSynchronized : BIT;
    Accuracy0             : BIT;
    Accuracy1             : BIT;
    Accuracy2             : BIT;
    Accuracy3             : BIT;
    Accuracy4             : BIT;
END_STRUCT
END_TYPE

```

Komponenten

| Name | Typ | Beschreibung |
|----------------------|-----|---|
| LeapSecondsKnown | BIT | Qualität der Schaltsekunden: FALSE := Unbekannt, TRUE := Bekannt. |
| ClockFailure | BIT | Qualität des Zeitgebers: FALSE := Gut, TRUE := Fehlerhaft. |
| ClockNotSynchronized | BIT | Qualität der Uhrzeitsynchronisierung: FALSE := Synchron, TRUE := Nicht Synchron. |
| Accuracy0 | BIT | Genauigkeit des Bruchteils der Sekunde. Dieser Wert bestimmt die max. Anzahl der signifikanten Bits des <code>T.UtcTime [▶ 479].fractionOfSecond-Members</code> . |
| Accuracy1 | BIT | |
| Accuracy2 | BIT | |
| Accuracy3 | BIT | |
| Accuracy4 | BIT | |

6.6.170 U_AcsiAnalogueCtlVal

Analogwert als UNION-Typ (Integer oder Fließkommazahl).

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
TYPE U_AcsiAnalogueCtlVal :
UNION
    i : DINT;
    f : REAL;
END_UNION
END_TYPE
```

Komponenten

| Name | Typ | Beschreibung |
|------|------|-------------------------------|
| i | DINT | Analogwert als Integerzahl |
| f | REAL | Analogwert als Fließkommazahl |

6.6.171 U_AcsiCtlVal

Allgemeiner Steuerwert für verschiedene Datenklassen (CDC), die bei der Schaltersteuerung verwendet werden.

Namensraum: [Tc3_Acsi \[▶ 96\]](#)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Syntax

```
TYPE U_AcsiCtlVal :
UNION
    SPC : BOOL;
    DPC : BOOL;
    INC : DINT;
    ENC : U_AcsiEnumeratedCtlVal;
    BSC : E_AcsiTcmd;
    ISC : SINT(-64..63);
    APC : U_AcsiAnalogueCtlVal;
    BAC : E_AcsiTcmd;
END_UNION
END_TYPE
```

Komponenten

| Name | Typ | Beschreibung |
|------|--|--|
| SPC | BOOL | Steuerwert für Einzelbefehl (Controllable Single Point):
FALSE := Off
TRUE := On |
| DPC | BOOL | Steuerwert für Doppelbefehl (Controllable Double Point):
FALSE := Off
TRUE := On |
| INC | DINT | Steuerwert für einen Sollwert (Controllable Integer Status). |
| ENC | U_AcsiEnumeratedCtIVal [▶ 482] | Steuerwert für einen Sollwert (Controllable Enumerated Status). |
| BSC | E_AcsiTcmd [▶ 437] | Steuerwert für Stufenstellbefehl (Binary Controlled Step Position Information). |
| ISC | SINT(-64..63) | Steuerwert für Stufenstellbefehl (Integer Controlled Step Position Information). |
| APC | U_AcsiAnalogueCtIVal [▶ 481] | Steuerwert für einen analogen Sollwert (Controllable Analog Process Value). |
| BAC | E_AcsiTcmd [▶ 437] | Steuerwert für einen analogen Sollwert (Binary Controlled Analog Process Value). |

6.6.172 U_AcsiEnumeratedCtIVal

Steuerwert für Controllable Enumerated Status (ENC).

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Bibliothek: [Tc3_Acsi](#) ([Tc3_Acsi.compiled-library](#))

Syntax

```

TYPE U_AcsiEnumeratedCtIVal :
UNION
  iAny      : SINT;
  Mode      : E_AcsiMod;
  PmpCtlGen : E_AcsiPmpCtlGen;
  PmpCtl    : E_AcsiPmpCtl;
  FanCtlGen : E_AcsiFanCtlGen;
  FanCtl    : E_AcsiFanCtl;
  OpModSyn  : E_AcsiOpModSyn;
END_UNION
END_TYPE

```

Komponenten

| Name | Typ | Beschreibung |
|-----------|---|---|
| iAny | SINT | Steuerwert als generischer 8-Bit-Integer. |
| Mode | E_AcsiMod [▶ 415] | Steuerwert für Modus. |
| PmpCtlGen | E_AcsiPmpCtlGen [▶ 423] | Generischer Steuerwert für Pumpenkontrolle. |
| PmpCtl | E_AcsiPmpCtl [▶ 422] | Steuerwert für Pumpenkontrolle. |
| FanCtlGen | E_AcsiFanCtlGen [▶ 409] | Generischer Steuerwert für Lüfterkontrolle. |
| FanCtl | E_AcsiFanCtl [▶ 408] | Steuerwert für Lüfterkontrolle. |
| OpModSyn | E_AcsiOpModSyn [▶ 418] | Steuerwert für den Operationsmodus. |

6.7 Parameterlisten

6.7.1 Param_Acsi

Namensraum: [Tc3_Acsi](#) [[▶ 96](#)]

Typ: Parameterliste

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Parameter

| Name | Typ | Wert | Beschreibung |
|------------------------------|-------------------|--|--|
| cMaxLengthOfObjectName | UDINT | 64 | Maximale Länge des Objektnamenstrings. |
| cMaxLengthOfObjectReference | UDINT | 255 | Maximale Länge des Referenzstrings. |
| cMaxLengthOfVisString129 | UDINT | 129 | Maximale Länge des „VisStr129“ Datentyps. |
| cMaxLengthOfClassname | UDINT | 20 | Maximale Namenslänge des Logischen-Knoten oder CDC's. |
| cDefault_TimeStamp | T_UtcTime | secondSinceEpoch:=DT#1970-01-01-00:00:00,
fractionOfSecond:=[0,0,0],
quality:=(LeapSecondsKnown:=0,
ClockFailure:=0,
ClockNotSynchronized:=0,
Accuracy0:=0,
Accuracy1:=0,
Accuracy2:=0,
Accuracy3:=0,
Accuracy4:=0) | Default-Initialwert für alle „TimeStamp“ DA-Instanzen. |
| cDefault_EntryTime_D | DATE | D#1984-01-01 | Default-Initialwert für alle „EntryTime.day“ Datenelement-Instanzen. |
| cDefault_EntryTime_TimeOfDay | TOD | TOD#0:0:0 | Default-Initialwert für alle „EntryTime.TimeOfDay“ Datenelement-Instanzen. |
| cDefault_EntryTime | T_BinaryTime | timeOfDay:=cDefault_EntryTime_TimeOfDay, day:=0 | Default-Initialwert für alle „EntryTime“ DA-Instanzen. |
| cDefault_Currency | STRING(3) | " | Default-Initialwert für alle „Currency“ DA-Instanzen. |
| cDefault_EntryID | T_OCTET8 | [8(0)] | Default-Initialwert für alle „EntryID“ DA-Instanzen. |
| cDefault_Originator | ST_AcsiOriginator | orCat:=E_AcsiOrCategory.NotSupported,
orIdnt:=[64(0)] | Default-Initialwert für all „Originator“ DA-Instanzen. |
| cDefault_Quality | ST_AcsiQuality | Validity0:=0,
Validity1:=0,
Overflow:=0,
OutOfRange:=0,
BadReference:=0,
Oscillatory:=0,
Failure:=0,
OldData:=0,
Inconsistent:=0,
Inaccurate:=0,
Source:=0, Test:=0,
OperatorBlocked:=0 | Default-Initialwert für alle „Quality“ DA-Instanzen. |
| cDefault_Dbpos | E_AcsiDbpos | E_AcsiDbpos.Intermediate | Default-Initialwert für alle „Dbpos“ DA-Instanzen. |
| cDefault_Tcmd | E_AcsiTcmd | E_AcsiTcmd.Stop | Default-Initialwert für alle „Tcmd“ DA-Instanzen. |

| Name | Typ | Wert | Beschreibung |
|----------------------------|--------------------------|--|--|
| cDefault_BrCB_OptionFields | ST_AcsiOptionalFields | SequenceNumber:=TRUE,
ReportTimeStamp:=TRUE,
ReasonForInclusion:=TRUE,
DataSetName:=TRUE,
DataReference:=FALSE,
BufferOverflow:=TRUE, EntryID:=TRUE,
ConfRevision:=TRUE,
Segmentation:=FALSE | Default-Initialwert für alle BRCB „OptionalFields“ DA-Instanzen. |
| cDefault_BrCB_TriggerOps | ST_AcsiTriggerConditions | DataChange:=FALSE,
QualityChange:=FALSE,
DataUpdate:=FALSE,
Integrity:=FALSE,
GeneralInterrogation:=TRUE | Default-Initialwert für alle BRCB „TriggerConditions“ DA-Instanzen. |
| cDefault_BrCB_TimeOfEntry | T_BinaryTime | timeOfDay:=TOD#0:0:0, day:=0 | Default-Initialwert für alle BRCB „TimeOfEntry“ DA-Instanzen. |
| cDefault_UrCB_OptionFields | ST_AcsiOptionalFields | SequenceNumber:=TRUE,
ReportTimeStamp:=TRUE,
ReasonForInclusion:=TRUE,
DataSetName:=TRUE,
DataReference:=FALSE,
BufferOverflow:=FALSE,
EntryID:=FALSE,
ConfRevision:=TRUE,
Segmentation:=FALSE | Default-Initialwert für alle URCB „OptionFields“ DA-Instanzen. |
| cDefault_UrCB_TriggerOps | ST_AcsiTriggerConditions | DataChange:=FALSE,
QualityChange:=FALSE,
DataUpdate:=FALSE,
Integrity:=FALSE,
GeneralInterrogation:=TRUE | Default-Initialwert für alle URCB „TriggerConditions“ DA-Instanzen. |
| cDefault_GoCB_MinTime | UDINT(1..16#7FFFFFFF) | 50 | Default GOCB Verzögerungszeit fürs Senden der Wiederholungen in Millisekunden. |
| cDefault_GoCB_MaxTime | UDINT(1..16#7FFFFFFF) | 5000 | Default GOCB Überwachungszeit in Millisekunden. |
| cDefault_GoCB_Multiplier | BYTE(1..BYTE#16#FF) | 2 | Default-Initialwert für GOOSE-Publisher Übertragungswiederholungs-Multiplikator. |
| cDefault_GoCB_Security | DWORD(0..16#0FFFFFFF) | 0 | Default GOOSE header Reserved1 and Reserved2 28 security bits. |

| Name | Typ | Wert | Beschreibung |
|-------------------------------------|-----------------------------|--|--|
| cDefault_LCB_OldEntrTm | T_BinaryTime | timeOfDay:=TOD#0:0:0, day:=0 | Default-Initialwert für alle LCB „OldEntrTm“. |
| cDefault_LCB_NewEntrTm | T_BinaryTime | timeOfDay:=TOD#0:0:0, day:=0 | Default-Initialwert für alle LCB „NewEntrTm“. |
| cDefault_LCB_OptFlds | ST_AcsiLogOptionalFields | ReasonForInclusion:=TRUE | Default-Initialwert für alle LCB „OptionFields“ DA-Instanzen. |
| cDefault_LCB_TrgOps | ST_AcsiTriggerConditions | DataChange:=FALSE, QualityChange:=FALSE, DataUpdate:=FALSE, Integrity:=FALSE, GeneralInterrogation:=TRUE | Default-Initialwert für alle LCB „TriggerConditions“ DA-Instanzen. |
| cDefault_MsvCB_OptFlds | ST_AcsiSvOptionalFields | Kein Bit gesetzt. | Default-Initialwert für alle MSVCB „OptionFields“ DA-Instanzen. |
| cDefault_UsvCB_OptFlds | ST_AcsiSvOptionalFields | Kein Bit gesetzt. | Default-Initialwert für alle USVCB „OptionFields“ DA-Instanzen. |
| cDefault_Per | E_AcsiPer | E_AcsiPer.Hourly | Default-Initialwert für alle „perRs“ oder „chrPerRs“ DA-Instanzen. |
| cDefault_Acs | E_AcsiAcs | E_AcsiAcs.Low | Default-Initialwert für alle „sptAcs“ DA-Instanzen. |
| | | | |
| cPresence_AnalogueValue | E_AcsiAnalogueValuePresCond | E_AcsiAnalogueValuePresCond.both_CO_f | Default-Präsenz der „AnalogueValue“-Datenelemente: „i“ und/oder „f“.
Im Statuswert sind „i“ und „f“ (beide) sichtbar, im Kontrollwert nur „f“ sichtbar. |
| cPresence_Cell_xEnd | BOOL | FALSE | Default-Präsenz von „Cell.xEnd“. |
| cPresence_Cell_yStart | BOOL | FALSE | Default-Präsenz von „Cell.yStart“. |
| cPresence_Cell_yEnd | BOOL | FALSE | Default-Präsenz von „Cell.yEnd“. |
| cPresence_Point_zVal | BOOL | FALSE | Default-Präsenz von „Point.zVal“. |
| cPresence_RangeConfig_limDb | BOOL | FALSE | Default-Präsenz von „RangeConfig.limDb“. |
| cPresence_Unit_multiplier | BOOL | FALSE | Default-Präsenz von „Unit.multiplier“. |
| cPresence_ValWithTrans_transInd | BOOL | FALSE | Default-Präsenz von „ValWithTrans.transInd“. |
| cPresence_Vector_ang | BOOL | FALSE | Default-Präsenz von „Vector.ang“. |
| cPresence_Oper_AnalogueValue_operTm | BOOL | FALSE | Default-Präsenz von „Oper.ctlVal(AnalogueValue).operTm“ |
| cPresence_Oper_BOOLEAN_operTm | BOOL | FALSE | Default-Präsenz von „Oper.ctlVal(BOOLEAN).operTm“ |
| cPresence_Oper_CODEDENUM_operTm | BOOL | FALSE | Default-Präsenz von „Oper.ctlVal(CODEDENUM).operTm“ |
| cPresence_Oper_ENUMERATED_operTm | BOOL | FALSE | Default-Präsenz von „Oper.ctlVal(ENUMERATED).operTm“ |

| Name | Typ | Wert | Beschreibung |
|---|------|-------|--|
| cPresence_Oper_I
NT32_operTm | BOOL | FALSE | Default-Präsenz von „Oper.ctlVal(INT32).operTm“ |
| cPresence_Oper_I
NT8_operTm | BOOL | FALSE | Default-Präsenz von „Oper.ctlVal(INT8).operTm“ |
| cPresence_SBOw_
AnalogueValue_op
erTm | BOOL | FALSE | Default-Präsenz von
„SBOw.ctlVal(AnalogueValue).operTm“ |
| cPresence_SBOw_
BOOLEAN_operT
m | BOOL | FALSE | Default-Präsenz von
„SBOw.ctlVal(BOOLEAN).operTm“ |
| cPresence_SBOw_
CODEDENUM_op
erTm | BOOL | FALSE | Default-Präsenz von
„SBOw.ctlVal(CODEDENUM).operTm“ |
| cPresence_SBOw_
ENUMERATED_op
erTm | BOOL | FALSE | Default-Präsenz von
„SBOw.ctlVal(ENUMERATED).operTm“ |
| cPresence_SBOw_
INT32_operTm | BOOL | FALSE | Default-Präsenz von
„SBOw.ctlVal(INT32).operTm“ |
| cPresence_SBOw_
INT8_operTm | BOOL | FALSE | Default-Präsenz von „SBOw.ctlVal(INT8).operTm“ |
| cPresence_Cancel_
AnalogueValue_o
perTm | BOOL | FALSE | Default-Präsenz von
„Cancel.ctlVal(AnalogueValue).operTm“ |
| cPresence_Cancel_
BOOLEAN_operT
m | BOOL | FALSE | Default-Präsenz von
„Cancel.ctlVal(BOOLEAN).operTm“ |
| cPresence_Cancel_
CODEDENUM_o
perTm | BOOL | FALSE | Default-Präsenz von
„Cancel.ctlVal(CODEDENUM).operTm“ |
| cPresence_Cancel_
ENUMERATED_o
perTm | BOOL | FALSE | Default-Präsenz von
„Cancel.ctlVal(ENUMERATED).operTm“ |
| cPresence_Cancel_
INT32_operTm | BOOL | FALSE | Default-Präsenz von
„Cancel.ctlVal(INT32).operTm“ |
| cPresence_Cancel_
INT8_operTm | BOOL | FALSE | Default-Präsenz von
„Cancel.ctlVal(INT8).operTm“ |
| cPresence_BrCB_
ResvTms | BOOL | FALSE | Default-Präsenz von „BRCB.ResvTms“ |
| cPresence_BrCB_
Owner | BOOL | FALSE | Default-Präsenz von „BRCB.Owner“ |
| cPresence_UrCB_
Owner | BOOL | FALSE | Default-Präsenz von „URCB.Owner“ |
| cPresence_LCB_O
ptFlds | BOOL | FALSE | Default-Präsenz von „LCB.OptFlds“ |
| cPresence_LCB_B
ufTm | BOOL | FALSE | Default-Präsenz von „LCB.BufTm“ |
| cPresence_GoCB_
MinTime | BOOL | FALSE | Default-Präsenz von „GOCB.MinTime“ |
| cPresence_GoCB_
MaxTime | BOOL | FALSE | Default-Präsenz von „GOCB.MaxTime“ |
| cPresence_GoCB_
FixedOffs | BOOL | FALSE | Default-Präsenz von „GOCB.FixedOffs“ |

6.7.2 Param_Gse

Namensraum: [Tc3_Gse \[► 97\]](#)

Typ: Parameterliste

Bibliothek: Tc3_Gse (Tc3_Gse.compiled-library)

Parameter

| Name | Typ | Wert | Beschreibung |
|----------------------------|-------------------------|-------|--|
| cMaxGseCtxBufferFra
mes | UDINT(1..256) | 32 | Max. Anzahl der gepufferten GSE-Telegramme (GOOSE, SV, GSSE) im Kontext-Puffer. |
| cMaxGseCtxBufferSiz
e | UDINT(1600..3200
00) | 32000 | Max. Bytegröße des Kontext-Puffers. Dieser Puffer dient der Speicherung der GSE-Telegramme (GOOSE, SV, GSSE), die in einem anderen Task-Kontext als im Kontext der IEC 61850 SPS-Task empfangen wurden. GSE-Telegramme können in einem anderen Task-Kontext empfangen werden, wenn mehrere Protokolle den gleichen Netzwerkadapter verwenden (z. B. GOOSE und EtherCAT Automation Protocol). Die gepufferten GSE-Telegramme werden immer von der IEC 61850 SPS-Task verarbeitet. |

6.7.3 Param_Sockets

Namensraum: [Tc3_Sockets \[► 97\]](#)

Typ: Parameterliste

Bibliothek: Tc3_Sockets (Tc3_Sockets.compiled-library)

Parameter

| Name | Typ | Wert | Beschreibung |
|-----------------------------------|---|--|--|
| cMaxSocketReceiveB
ufferSize | UDINT | 16#20000 | Max. Bytegröße des Empfangspuffers für die Client/Server-Telegramme. |
| cDefaultReceiverThrot
tleTimes | T_ThrottleTime
s [► 453] | [T#0S, T#10MS,
T#20MS, T#40MS,
T#60MS, T#80MS,
T#100MS,
T#120MS,
T#140MS,
T#160MS,
T#180MS,
T#200MS,
T#250MS] | Default Zeiten zur Drosselung der Abfragezyklen/Abrufhäufigkeit der empfangenen TCP/IP-Telegramme (Client/Server-Kommunikation). |

6.8 Globale Variablen

6.8.1 TC_EVENTS Konstanten

TC_EVENTS ist eine globale TwinCAT System Variablenliste (GVL). Jede Konstante in dieser Liste repräsentiert eine auf einem TwinCAT System verfügbare Ereignisklasse. Die Instanzen der Ereignisklassen sind strukturierte Datentypen. Deren Membervariablen sind Instanzen einzelner Ereignisse. Die Ereignisse können z. B. in einem SPS-Projekt oder einer SPS-Bibliothek zum Loggen unterschiedlichster Meldungen verwendet werden. Das TwinCAT System besitzt einige vordefinierte Ereignisklassen. Weitere Ereignisklassen werden von den eingebundenen SPS-Bibliotheken (Functions) mitgebracht, oder vom Anwender in der SPS-Applikation definiert. Die Globale Variablenliste der Ereignisklassen wird vom TwinCAT System automatisch erstellt und aktualisiert. Die Anzahl der verfügbaren Ereignisklassen kann dadurch von System zu System variieren.

Namensraum: [TwinCAT TypeSystem \[▶ 96\].TC_EVENTS](#)

Typ: Globale Variablenliste (GVL)

Bibliothek: Nicht erforderlich. TwinCAT System Resource ab v3.1.4022.14

TwinCAT System Ereignisklassen

| Name | Typ |
|------------------------|---|
| TcSystemEventClass | ST_TcSystemEventClass [▶ 512] |
| TcGeneralAdsEventClass | ST_TcGeneralAdsEventClass [▶ 494] |
| TcRouterEventClass | ST_TcRouterEventClass [▶ 505] |
| TcRTimeEventClass | ST_TcRTimeEventClass [▶ 506] |
| Win32EventClass | ST_Win32EventClass [▶ 519] |

TF6510 Ereignisklassen

| Name | Typ |
|---------------------------------------|--|
| TcIEC61850EventClass | ST TcIEC61850EventClass [▶ 498] |
| TcScsmEventClass | ST TcScsmEventClass [▶ 509] |
| | |
| TcMmsAbortEventClass | ST TcMmsAbortEventClass [▶ 498] |
| TcMmsAssociationEventClass | ST TcMmsAssociationEventClass [▶ 498] |
| TcMmsClientEventClass | ST TcMmsClientEventClass [▶ 499] |
| TcMmsIncomingConnectEventClass | ST TcMmsIncomingConnectEventClass [▶ 499] |
| TcMmsIncomingDisconnectEventClass | ST TcMmsIncomingDisconnectEventClass [▶ 499] |
| TcMmsOutgoingConnectEventClass | ST TcMmsOutgoingConnectEventClass [▶ 500] |
| TcMmsOutgoingDisconnectEventClass | ST TcMmsOutgoingDisconnectEventClass [▶ 500] |
| TcMmsReceiverEventClass | ST TcMmsReceiverEventClass [▶ 501] |
| TcMmsSenderEventClass | ST TcMmsSenderEventClass [▶ 501] |
| TcMmsTransactionEventClass | ST TcMmsTransactionEventClass [▶ 501] |
| | |
| TcUlosiAbortEventClass | ST TcUlosiAbortEventClass [▶ 515] |
| TcUlosiAssociationEventClass | ST TcUlosiAssociationEventClass [▶ 516] |
| TcUlosiClientEventClass | ST TcUlosiClientEventClass [▶ 516] |
| TcUlosiIncomingConnectEventClass | ST TcUlosiIncomingConnectEventClass [▶ 516] |
| TcUlosiIncomingDisconnectEventClass | ST TcUlosiIncomingDisconnectEventClass [▶ 517] |
| TcUlosiOutgoingConnectEventClass | ST TcUlosiOutgoingConnectEventClass [▶ 517] |
| TcUlosiOutgoingDisconnectEventClass | ST TcUlosiOutgoingDisconnectEventClass [▶ 518] |
| TcUlosiReceiverEventClass | ST TcUlosiReceiverEventClass [▶ 518] |
| TcUlosiSenderEventClass | ST TcUlosiSenderEventClass [▶ 518] |
| TcUlosiTransactionEventClass | ST TcUlosiTransactionEventClass [▶ 518] |
| | |
| TcRfc1006AbortEventClass | ST TcRfc1006AbortEventClass [▶ 502] |
| TcRfc1006AssociationEventClass | ST TcRfc1006AssociationEventClass [▶ 502] |
| TcRfc1006ClientEventClass | ST TcRfc1006ClientEventClass [▶ 502] |
| TcRfc1006IncomingConnectEventClass | ST TcRfc1006IncomingConnectEventClass [▶ 503] |
| TcRfc1006IncomingDisconnectEventClass | ST TcRfc1006IncomingDisconnectEventClass [▶ 503] |
| TcRfc1006OutgoingConnectEventClass | ST TcRfc1006OutgoingConnectEventClass [▶ 503] |
| TcRfc1006OutgoingDisconnectEventClass | ST TcRfc1006OutgoingDisconnectEventClass [▶ 504] |
| TcRfc1006ReceiverEventClass | ST TcRfc1006ReceiverEventClass [▶ 504] |
| TcRfc1006SenderEventClass | ST TcRfc1006SenderEventClass [▶ 504] |
| TcRfc1006TransactionEventClass | ST TcRfc1006TransactionEventClass [▶ 505] |
| | |
| TcTpktAbortEventClass | ST TcTpktAbortEventClass [▶ 514] |
| TcTpktAssociationEventClass | ST TcTpktAssociationEventClass [▶ 514] |
| TcTpktPduEventClass | ST TcTpktPduEventClass [▶ 514] |
| TcTpktReceiverEventClass | ST TcTpktReceiverEventClass [▶ 514] |
| TcTpktSenderEventClass | ST TcTpktSenderEventClass [▶ 515] |
| TcTpktTransactionEventClass | ST TcTpktTransactionEventClass [▶ 515] |
| | |
| TcSocketsAbortEventClass | ST TcSocketsAbortEventClass [▶ 509] |

| Name | Typ |
|---------------------------------------|--|
| TcSocketsAssociationEventClass | ST_TcSocketsAssociationEventClass [▶ 509] |
| TcSocketsClientEventClass | ST_TcSocketsClientEventClass [▶ 510] |
| TcSocketsIncomingConnectEventClass | ST_TcSocketsIncomingConnectEventClass [▶ 510] |
| TcSocketsIncomingDisconnectEventClass | ST_TcSocketsIncomingDisconnectEventClass [▶ 510] |
| TcSocketsOutgoingConnectEventClass | ST_TcSocketsOutgoingConnectEventClass [▶ 511] |
| TcSocketsOutgoingDisconnectEventClass | ST_TcSocketsOutgoingDisconnectEventClass [▶ 511] |
| TcSocketsReceiverEventClass | ST_TcSocketsReceiverEventClass [▶ 511] |
| TcSocketsSenderEventClass | ST_TcSocketsSenderEventClass [▶ 512] |
| TcSocketsTransactionEventClass | ST_TcSocketsTransactionEventClass [▶ 512] |
| TcSocketsWin32SrvEventClass | ST_TcSocketsWin32SrvEventClass [▶ 512] |
| | |
| TcAcsiEventClass | ST_TcAcsiEventClass [▶ 493] |

6.8.1.1 ST_TcAcsiEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der ACSI-Ereignisklasse.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcAcsiEventClass

Typ: Globale Variable

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

```

TYPE ST_TcAcsiEventClass:
STRUCT
    NoError : TcEventEntry;
    InstanceNotAvailable : TcEventEntry;
    InstanceInUse : TcEventEntry;
    AccessViolation : TcEventEntry;
    AccessNotAllowedInCurrentState : TcEventEntry;
    ParameterValueInappropriate : TcEventEntry;
    ParameterValueInconsistent : TcEventEntry;
    ClassNotSupported : TcEventEntry;
    InstanceLockedByOtherClient : TcEventEntry;
    ControlMustBeSelected : TcEventEntry;
    TypeConflict : TcEventEntry;
    FailedDueToCommunicationsConstraint : TcEventEntry;
    FailedDueToServerConstraint : TcEventEntry;
END_STRUCT
END_TYPE
    
```

Ereignisse

| Name | ID | Severity | Beschreibung |
|-------------------------------------|----|----------|--------------|
| NoError | 0 | Verbose | |
| InstanceNotAvailable | 1 | Error | |
| InstanceInUse | 2 | Error | |
| AccessViolation | 3 | Error | |
| AccessNotAllowedInCurrentState | 4 | Error | |
| ParameterValueInappropriate | 5 | Error | |
| ParameterValueInconsistent | 6 | Error | |
| ClassNotSupported | 7 | Error | |
| InstanceLockedByOtherClient | 8 | Error | |
| ControlMustBeSelected | 9 | Error | |
| TypeConflict | 10 | Error | |
| FailedDueToCommunicationsConstraint | 11 | Error | |
| FailedDueToServerConstraint | 12 | Error | |

6.8.1.2 ST_TcGeneralAdsEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der TwinCAT ADS Ereignisklasse.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcGeneralAdsEventClass

Typ: Globale Variable

Bibliothek: Nicht erforderlich, TwinCAT System Resource ab v3.1.4022.14

Ereignisse

| Name | ID | | Severity | Beschreibung |
|---------------------------|-------|------|----------|---|
| | Hex | Dez | | |
| ServiceNotSupported | 0x701 | 1793 | Error | Service wird vom Server nicht unterstützt |
| InvalidIndexGroup | 0x702 | 1794 | Error | Ungültige Index-Gruppe |
| InvalidIndexOffset | 0x703 | 1795 | Error | Ungültiger Index-Offset |
| InvalidAccess | 0x704 | 1796 | Error | Lesen und schreiben nicht gestattet. |
| InvalidSize | 0x705 | 1797 | Error | Parametergröße nicht korrekt |
| InvalidData | 0x706 | 1798 | Error | Ungültige Parameter-Werte |
| NotReady | 0x707 | 1799 | Error | Gerät ist nicht betriebsbereit |
| Busy | 0x708 | 1800 | Error | Gerät ist beschäftigt |
| InvalidContext | 0x709 | 1801 | Error | Ungültiger Kontext (muss in Windows sein) |
| NoMemory | 0x70A | 1802 | Error | Nicht genügend Speicher |
| InvalidParam | 0x70B | 1803 | Error | Ungültige Parameter-Werte |
| NotFound | 0x70C | 1804 | Error | Nicht gefunden (Dateien,...) |
| Syntax | 0x70D | 1805 | Error | Syntax-Fehler in Datei oder Befehl |
| Incompatible | 0x70E | 1806 | Error | Objekte stimmen nicht überein |
| AlreadyExists | 0x70F | 1807 | Error | Objekt ist bereits vorhanden |
| SymbolNotFound | 0x710 | 1808 | Error | Symbol nicht gefunden |
| SymbolVersionInvalid | 0x711 | 1809 | Error | Symbol-Version ungültig |
| InvalidState | 0x712 | 1810 | Error | Gerät im ungültigen Zustand |
| TransModeNotSupported | 0x713 | 1811 | Error | AdsTransMode nicht unterstützt |
| NotificationHandleInvalid | 0x714 | 1812 | Error | Notification Handle ist ungültig |
| ClientUnknown | 0x715 | 1813 | Error | Notification-Client nicht registriert |
| NoMoreHandles | 0x716 | 1814 | Error | Keine weitere Notification Handle |
| InvalidWatchSize | 0x717 | 1815 | Error | Größe der Notification zu groß |
| NotInit | 0x718 | 1816 | Error | Gerät nicht initialisiert |
| DeviceTimeout | 0x719 | 1817 | Error | Gerät hat einen Timeout |
| NoInterface | 0x71A | 1818 | Error | Interface Abfrage fehlgeschlagen |

| Name | ID | | Severity | Beschreibung |
|-------------------------|-------|------|----------|---|
| | Hex | Dez | | |
| InvalidInterface | 0x71B | 1819 | Error | Falsches Interface angefordert |
| InvalidClassID | 0x71C | 1820 | Error | Class-ID ist ungültig |
| InvalidObjectID | 0x71D | 1821 | Error | Object-ID ist ungültig |
| Pending | 0x71E | 1822 | Error | Anforderung steht aus |
| Aborted | 0x71F | 1823 | Error | Anforderung wird abgebrochen |
| SignalWarning | 0x720 | 1824 | Error | Signal-Warnung |
| InvalidArrayIndex | 0x721 | 1825 | Error | Ungültiger Array-Index |
| SymbolNotActive | 0x722 | 1826 | Error | Symbol nicht aktiv |
| AccessDenied | 0x723 | 1827 | Error | Zugriff verweigert |
| LicenseMissing | 0x724 | 1828 | Error | Fehlende Lizenz |
| LicenseExpired | 0x725 | 1829 | Error | Lizenz abgelaufen |
| LicenseExceeded | 0x726 | 1830 | Error | Lizenz überschritten |
| LicenseInvalid | 0x727 | 1831 | Error | Lizenz ungültig |
| LicenseSystemID | 0x728 | 1832 | Error | Lizenz der System-ID ungültig |
| LicenseNoTimeLimit | 0x729 | 1833 | Error | Lizenz nicht zeitlich begrenzt |
| LicenseFutureIssue | 0x72A | 1834 | Error | Lizenzproblem: Zeitpunkt in der Zukunft |
| LicenseTimeTooLong | 0x72B | 1835 | Error | Lizenz-Zeitraum zu lang |
| Exception | 0x72c | 1836 | Error | Exception beim Systemstart |
| LicenseDuplicated | 0x72D | 1837 | Error | Lizenz-Datei zweimal gelesen |
| | 0x72E | 1838 | Error | Ungültige Signatur |
| | 0x72F | 1839 | Error | öffentliches Zertifikat |
| | 0x740 | 1856 | Error | Clientfehler |
| InvalidServiceParam | 0x741 | 1857 | Error | Dienst enthält einen ungültigen Parameter |
| ListEmpty | 0x742 | 1858 | Error | Polling-Liste ist leer |
| VariableConAlreadyInUse | 0x743 | 1859 | Error | Var-Verbindung bereits im Einsatz |
| InvokelDAAlreadyInUse | 0x744 | 1860 | Error | Die aufgerufene ID ist bereits in Benutzung |
| TimeoutElapsed | 0x745 | 1861 | Error | Timeout ist aufgetreten |
| Win32Error | 0x746 | 1862 | Error | Fehler im Win32 Subsystem |
| TimeoutInvalid | 0x747 | 1863 | Error | Ungültiger Client Timeout-Wert |
| PortNotOpen | 0x748 | 1864 | Error | ADS-Port nicht geöffnet |
| NoAMSAddr | 0x749 | 1865 | Error | |

| Name | ID | | Severity | Beschreibung |
|-------------------|-------|------|----------|----------------------------------|
| | Hex | Dez | | |
| SyncInternalError | 0x750 | 1872 | Error | Interner Fehler in Ads-Sync |
| AddHash | 0x751 | 1873 | Error | Hash-Tabelle-Überlauf |
| RemoveHash | 0x752 | 1874 | Error | Schlüssel nicht gefunden im Hash |
| NoMoreSymbols | 0x753 | 1875 | Error | Keine weitere Symbole im Cache |
| SyncResInvalid | 0x754 | 1876 | Error | Ungültige Antwort empfangen |
| SyncPortLocked | 0x755 | 1877 | Error | Sync Port ist gesperrt |

6.8.1.3 ST_TcIEC61850EventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der IEC 61850-Ereignisklasse.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcIEC61850EventClass

Typ: Globale Variable

Bibliothek: Tc3_iec61850 (Tc3_iec61850.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidParameterValue | 1 | Error | |
| InvalidObjectState | 2 | Error | |
| InvalidMmsObject | 3 | Error | |
| InvalidAssociationObject | 4 | Error | |
| ObjectNotFound | 5 | Error | |
| InvalidIEDObject | 6 | Error | |

6.8.1.4 ST_TcMmsAbortEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der MMS-Ereignisklasse für den Verbindungsabbruch.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcMmsAbortEventClass

Typ: Globale Variable

Bibliothek: Tc3_Mms (Tc3_Mms.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidAssociationObject | 1 | Error | |
| InvalidUlosiObject | 2 | Error | |

6.8.1.5 ST_TcMmsAssociationEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der MMS-Ereignisklasse für die Verwaltung einer Verbindung (Association).

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcMmsAssociationEventClass

Typ: Globale Variable

Bibliothek: Tc3_Mms (Tc3_Mms.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| StateTransitionError | 1 | Error | |
| StateValidationError | 2 | Error | |
| DecoderFailure | 3 | Error | |
| EncoderFailure | 4 | Error | |
| InvalidAssociationObject | 5 | Error | |
| InvalidPduObject | 6 | Error | |

6.8.1.6 ST_TcMmsClientEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der MMS-Ereignisklasse für die Verwaltung einer Clientinstanz.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcMmsClientEventClass

Typ: Globale Variable

Bibliothek: Tc3_Mms (Tc3_Mms.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidAssociationObject | 1 | Error | |

6.8.1.7 ST_TcMmsIncomingConnectEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der MMS-Ereignisklasse für den eingehenden Verbindungsaufbau.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcMmsIncomingConnectEventClass

Typ: Globale Variable

Bibliothek: Tc3_Mms (Tc3_Mms.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidPrimitiveState | 1 | Error | |
| InvalidAssociationObject | 2 | Error | |
| InvalidConnectionObject | 3 | Error | |
| InvalidSettingsObject | 4 | Error | |
| InvalidUlosiObject | 5 | Error | |
| InvalidServerObject | 6 | Error | |
| InitiateIndTimeoutError | 7 | Error | |
| InitiateRspTimeoutError | 8 | Error | |
| PduCreationError | 9 | Error | |

6.8.1.8 ST_TcMmsIncomingDisconnectEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der MMS-Ereignisklasse für den eingehenden Verbindungsabbau.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcMmsIncomingDisconnectEventClass

Typ: Globale Variable

Bibliothek: Tc3_Mms (Tc3_Mms.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidPrimitiveState | 1 | Error | |
| InvalidAssociationObject | 2 | Error | |
| InvalidConnectionObject | 3 | Error | |
| InvalidUlosiObject | 4 | Error | |
| InvalidClientObject | 5 | Error | |
| ConcludeRspTimeoutError | 6 | Error | |
| PduCreationError | 7 | Error | |

6.8.1.9 ST_TcMmsOutgoingConnectEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der MMS-Ereignisklasse für den ausgehenden Verbindungsaufbau.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcMmsOutgoingConnectEventClass

Typ: Globale Variable

Bibliothek: Tc3_Mms (Tc3_Mms.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidPrimitiveState | 1 | Error | |
| InvalidAssociationObject | 2 | Error | |
| InvalidClientObject | 3 | Error | |
| InvalidSettingsObject | 4 | Error | |
| InvalidUlosiObject | 5 | Error | |
| InitiateCnfTimeoutError | 6 | Error | |
| PduCreationError | 7 | Error | |

6.8.1.10 ST_TcMmsOutgoingDisconnectEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der MMS-Ereignisklasse für den ausgehenden Verbindungsabbau.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcMmsOutgoingDisconnectEventClass

Typ: Globale Variable

Bibliothek: Tc3_Mms (Tc3_Mms.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidPrimitiveState | 1 | Error | |
| InvalidAssociationObject | 2 | Error | |
| InvalidConnectionObject | 3 | Error | |
| InvalidSettingsObject | 4 | Error | |
| InvalidUlosiObject | 5 | Error | |
| InvalidClientObject | 6 | Error | |
| ConcludeCnfTimeoutError | 7 | Error | |
| PduCreationError | 8 | Error | |

6.8.1.11 ST_TcMmsReceiverEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der MMS-Ereignisklasse für den Empfang der Daten.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcMmsReceiverEventClass

Typ: Globale Variable

Bibliothek: Tc3_Mms (Tc3_Mms.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidAssociationObject | 1 | Error | |
| InvalidSettingsObject | 2 | Error | |
| InvalidUlosiObject | 3 | Error | |
| InvalidPrimitiveState | 4 | Error | |
| ResponseTimeoutError | 5 | Error | |
| DecoderFailure | 6 | Error | |

6.8.1.12 ST_TcMmsSenderEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der MMS-Ereignisklasse für den Versand der Daten.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcMmsSenderEventClass

Typ: Globale Variable

Bibliothek: Tc3_Mms (Tc3_Mms.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|-----------------------|----|----------|--------------|
| InvalidPrimitiveState | 1 | Error | |

6.8.1.13 ST_TcMmsTransactionEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der MMS-Ereignisklasse für die Verwaltung der Datentransaktionen.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcMmsTransactionEventClass

Typ: Globale Variable

Bibliothek: Tc3_Mms (Tc3_Mms.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidAssociationObject | 1 | Error | |
| InvalidSettingsObject | 2 | Error | |
| InvalidUlosiObject | 3 | Error | |
| EncoderFailure | 4 | Error | |
| TransactionTimeoutError | 5 | Error | |

6.8.1.14 ST_TcRfc1006AbortEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der RFC 1006-Ereignisklasse für den Verbindungsabbruch.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcRfc1006AbortEventClass

Typ: Globale Variable

Bibliothek: Tc3_Rfc1006 (Tc3_Rfc1006.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidAssociationObject | 1 | Error | |
| InvalidTpktObject | 2 | Error | |

6.8.1.15 ST_TcRfc1006AssociationEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der RFC 1006-Ereignisklasse für die Verwaltung einer Verbindung (Association).

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcRfc1006AssociationEventClass

Typ: Globale Variable

Bibliothek: Tc3_Rfc1006 (Tc3_Rfc1006.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|----------------------|----|----------|--------------|
| StateTransitionError | 1 | Error | |
| StateValidationError | 2 | Error | |

6.8.1.16 ST_TcRfc1006ClientEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der RFC 1006-Ereignisklasse für die Verwaltung einer Clientinstanz.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcRfc1006ClientEventClass

Typ: Globale Variable

Bibliothek: Tc3_Rfc1006 (Tc3_Rfc1006.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidAssociationObject | 1 | Error | |

6.8.1.17 ST_TcRfc1006IncomingConnectEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der RFC 1006-Ereignisklasse für den eingehenden Verbindungsaufbau.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcRfc1006IncomingConnectEventClass

Typ: Globale Variable

Bibliothek: Tc3_Rfc1006 (Tc3_Rfc1006.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidPrimitiveState | 1 | Error | |
| InvalidAssociationObject | 2 | Error | |
| InvalidSettingsObject | 3 | Error | |
| InvalidTpktObject | 4 | Error | |
| InvalidConnectionObject | 5 | Error | |
| InvalidServerObject | 6 | Error | |
| CRTimeoutError | 7 | Error | |
| CCTimeoutError | 8 | Error | |

6.8.1.18 ST_TcRfc1006IncomingDisconnectEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der RFC 1006-Ereignisklasse für den eingehenden Verbindungsabbau.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcRfc1006IncomingDisconnectEventClass

Typ: Globale Variable

Bibliothek: Tc3_Rfc1006 (Tc3_Rfc1006.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidPrimitiveState | 1 | Error | |
| InvalidAssociationObject | 2 | Error | |
| InvalidSettingsObject | 3 | Error | |
| InvalidTpktObject | 4 | Error | |
| DRTimeoutError | 5 | Error | |

6.8.1.19 ST_TcRfc1006OutgoingConnectEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der RFC 1006-Ereignisklasse für den ausgehenden Verbindungsaufbau.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcRfc1006OutgoingConnectEventClass

Typ: Globale Variable

Bibliothek: Tc3_Rfc1006 (Tc3_Rfc1006.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidPrimitiveState | 1 | Error | |
| InvalidAssociationObject | 2 | Error | |
| InvalidSettingsObject | 3 | Error | |
| InvalidTpktObject | 4 | Error | |
| InvalidClientObject | 5 | Error | |
| CCTimeoutError | 6 | Error | |

6.8.1.20 ST_TcRfc1006OutgoingDisconnectEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der RFC 1006-Ereignisklasse für den ausgehenden Verbindungsabbau.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcRfc1006OutgoingDisconnectEventClass

Typ: Globale Variable

Bibliothek: Tc3_Rfc1006 (Tc3_Rfc1006.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidPrimitiveState | 1 | Error | |
| InvalidAssociationObject | 2 | Error | |
| InvalidTpktObject | 3 | Error | |

6.8.1.21 ST_TcRfc1006ReceiverEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der RFC 1006-Ereignisklasse für den Empfang der Daten.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcRfc1006ReceiverEventClass

Typ: Globale Variable

Bibliothek: Tc3_Rfc1006 (Tc3_Rfc1006.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidAssociationObject | 1 | Error | |
| InvalidSettingsObject | 2 | Error | |
| InvalidTpktObject | 3 | Error | |
| InvalidPrimitiveState | 4 | Error | |
| DataRspTimeoutError | 5 | Error | |
| DecoderFailure | 6 | Error | |

6.8.1.22 ST_TcRfc1006SenderEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der RFC 1006-Ereignisklasse für den Versand der Daten.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcRfc1006SenderEventClass

Typ: Globale Variable

Bibliothek: Tc3_Rfc1006 (Tc3_Rfc1006.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|-----------------------|----|----------|--------------|
| InvalidPrimitiveState | 1 | Error | |
| DataCnfTimeoutError | 2 | Error | |

6.8.1.23 ST_TcRfc1006TransactionEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der RFC 1006-Ereignisklasse für die Verwaltung der Datentransaktionen.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcRfc1006TransactionEventClass

Typ: Globale Variable

Bibliothek: Tc3_Rfc1006 (Tc3_Rfc1006.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidAssociationObject | 1 | Error | |
| InvalidTpktObject | 2 | Error | |
| EncoderFailure | 3 | Error | |

6.8.1.24 ST_TcRouterEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der TwinCAT Router Ereignisklasse.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcRouterEventClass

Typ: Globale Variable

Bibliothek: Nicht erforderlich, TwinCAT System Resource ab v3.1.4022.14

Ereignisse

| Name | ID | | Severity | Beschreibung |
|------------------|-------|------|----------|---|
| | Hex | Dez | | |
| NoLockedMemory | 0x500 | 1280 | Error | Lockierter Speicher kann nicht zugewiesen werden. |
| ResizeMemory | 0x501 | 1281 | Error | Die Größe des Router-Speichers konnte nicht geändert werden. |
| MailboxFull | 0x502 | 1282 | Error | Das Postfach hat die maximale Anzahl der möglichen Meldungen erreicht. Die aktuell gesendete Nachricht wurde abgewiesen. |
| DebugBoxFull | 0x503 | 1283 | Error | Das Postfach hat die maximale Anzahl der möglichen Meldungen erreicht. Die gesendete Nachricht wird nicht im ADS Monitor angezeigt. |
| UnknownPortType | 0x504 | 1284 | Error | Der Port-Typ ist unbekannt. |
| NotInitialized | 0x505 | 1285 | Error | Router ist nicht initialisiert. |
| PortAlreadyInUse | 0x506 | 1286 | Error | Die gewünschte Port-Nummer ist bereits vergeben. |
| NotRegistered | 0x507 | 1287 | Error | Der Port ist nicht registriert. |
| NoMoreQueues | 0x508 | 1288 | Error | Die maximale Anzahl von Ports ist erreicht. |
| InvalidPort | 0x509 | 1289 | Error | Der Port ist ungültig. |
| NotActivated | 0x50A | 1290 | Error | Der Router ist nicht aktiv. |
| FragmentBoxFull | 0x50B | 1291 | Error | |
| FragmentTimeout | 0x50C | 1292 | Error | |
| ToBeRemoved | 0x50D | 1293 | Error | |

6.8.1.25 ST_TcRTimeEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der TwinCAT RTime Ereignisklasse.

Namensraum: `TC_EVENTS` [[▶ 489](#)].TcRTimeEventClass

Typ: Globale Variable

Bibliothek: Nicht erforderlich, TwinCAT System Resource ab v3.1.4022.14

Ereignisse

| Name | ID | | Severity | Beschreibung |
|---------------------|--------|------|----------|---|
| | Hex | Dez | | |
| InternalError | 0x1000 | 4096 | Error | Interner Fehler im TwinCAT Echtzeit-System. |
| BadTimerPeriods | 0x1001 | 4097 | Error | Timer-Wert ist nicht gültig. |
| InvalidTaskPtr | 0x1002 | 4098 | Error | Task-Pointer hat den ungültigen Wert 0 (null). |
| InvalidStackPtr | 0x1003 | 4099 | Error | Task Stackpointer hat den ungültigen Wert 0. |
| PrioExists | 0x1004 | 4100 | Error | Die Request Task Priority ist bereits vergeben. |
| NoMoreTCB | 0x1005 | 4101 | Error | Kein freies TCB (Task Control Block) zur Verfügung. Maximale Anzahl von TCBs beträgt 64. |
| NoMoreSemas | 0x1006 | 4102 | Error | Keine freien Semaphoren zur Verfügung. Maximale Anzahl der Semaphoren beträgt 64. |
| NoMoreQueues | 0x1007 | 4103 | Error | Kein freier Platz in der Warteschlange zur Verfügung. Maximale Anzahl der Plätze in der Warteschlange beträgt 64. |
| ExtIRQAlreadyDef | 0x100D | 4109 | Error | Ein externer Synchronisations-Interrupt wird bereits angewandt. |
| ExtIRQNotDef | 0x100E | 4110 | Error | Kein externer Synchronisations-Interrupt angewandt. |
| ExtIRQInstallFailed | 0x100F | 4111 | Error | Anwendung des externen Synchronisierungs-Interrupts ist fehlgeschlagen |
| IRQNotLessOrEqual | 0x1010 | 4112 | Error | Aufruf einer Service-Funktion im falschen Kontext |
| VMXNotSupported | 0x1017 | 4119 | Error | Intel VT-x Erweiterung wird nicht unterstützt. |
| VMXDisabled | 0x1018 | 4120 | Error | Intel VT-x Erweiterung ist nicht aktiviert im BIOS. |

| Name | ID | | Severity | Beschreibung |
|--------------------|--------|------|----------|--|
| | Hex | Dez | | |
| VMXControlsMissing | 0x1019 | 4121 | Error | Fehlende Funktion in Intel VT-x Erweiterung. |
| VMXEnableFails | 0x101A | 4122 | Error | Aktivieren von Intel VT-x schlägt fehl. |

6.8.1.26 ST_TcScsmEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der SCSM-Ereignisklasse.

Namensraum: TC_EVENTS [[▶ 489](#)].TcScsmEventClass

Typ: Globale Variable

Bibliothek: Tc3_iec61850_8_1 (Tc3_iec61850_8_1.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|---------------------------|----|----------|--------------|
| ConfirmedErrorPduReceived | 1 | Error | |
| RejectPduReceived | 2 | Error | |
| ReqAndCnfInvokeIDiffers | 3 | Error | |
| ConfirmationTimeout | 4 | Error | |
| InvalidObjectReference | 5 | Error | |
| InvalidAssociationObject | 6 | Error | |
| InvalidSettingsObject | 7 | Error | |
| InvalidMmsObject | 8 | Error | |
| InvalidClientObject | 9 | Error | |
| InvalidServerObject | 10 | Error | |
| InvalidConnectionObject | 11 | Error | |
| InvalidIEDObject | 12 | Error | |

6.8.1.27 ST_TcSocketsAbortEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der Socket-Ereignisklasse für den Verbindungsabbruch.

Namensraum: TC_EVENTS [[▶ 489](#)].TcSocketsAbortEventClass

Typ: Globale Variable

Bibliothek: Tc3_Sockets (Tc3_Sockets.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidAssociationObject | 1 | Error | |
| InvalidTcpObject | 2 | Error | |

6.8.1.28 ST_TcSocketsAssociationEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der Socket-Ereignisklasse für die Verwaltung einer Verbindungs-Assoziation

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcSocketsAssociationEventClass

Typ: Globale Variable

Bibliothek: Tc3_Sockets (Tc3_Sockets.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|-------------------------|----|----------|--------------|
| InvalidStateTransition | 1 | Error | |
| InvalidEnvironmentState | 2 | Error | |

6.8.1.29 ST_TcSocketsClientEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der Socket-Ereignisklasse für die Verwaltung einer Clientinstanz.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcSocketsClientEventClass

Typ: Globale Variable

Bibliothek: Tc3_Sockets (Tc3_Sockets.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidAssociationObject | 1 | Error | |

6.8.1.30 ST_TcSocketsIncomingConnectEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der Socket-Ereignisklasse für den eingehenden Verbindungsaufbau.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcSocketsIncomingConnectEventClass

Typ: Globale Variable

Bibliothek: Tc3_Sockets (Tc3_Sockets.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidPrimitiveState | 1 | Error | |
| InvalidAssociationObject | 2 | Error | |
| InvalidTcpObject | 3 | Error | |
| InvalidConnectionObject | 4 | Error | |
| InvalidServerObject | 5 | Error | |
| InvalidSettingsObject | 6 | Error | |
| ConnectRspTimeoutError | 7 | Error | |

6.8.1.31 ST_TcSocketsIncomingDisconnectEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der Socket-Ereignisklasse für den eingehenden Verbindungsabbau.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcSocketsIncomingDisconnectEventClass

Typ: Globale Variable

Bibliothek: Tc3_Sockets (Tc3_Sockets.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|---------------------------|----|----------|--------------|
| InvalidPrimitiveState | 1 | Error | |
| InvalidAssociationObject | 2 | Error | |
| InvalidTcpObject | 3 | Error | |
| DisconnectRspTimeoutError | 4 | Error | |

6.8.1.32 ST_TcSocketsOutgoingConnectEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der Socket-Ereignisklasse für den ausgehenden Verbindungsaufbau.

Namensraum: [TC_EVENTS](#) [▶ 489].TcSocketsOutgoingConnectEventClass

Typ: Globale Variable

Bibliothek: Tc3_Sockets (Tc3_Sockets.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidPrimitiveState | 1 | Error | |
| InvalidAssociationObject | 2 | Error | |
| InvalidClientObject | 3 | Error | |
| InvalidSettingsObject | 4 | Error | |
| InvalidTcpObject | 5 | Error | |

6.8.1.33 ST_TcSocketsOutgoingDisconnectEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der Socket-Ereignisklasse für den ausgehenden Verbindungsabbau.

Namensraum: [TC_EVENTS](#) [▶ 489].TcSocketsOutgoingDisconnectEventClass

Typ: Globale Variable

Bibliothek: Tc3_Sockets (Tc3_Sockets.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidPrimitiveState | 1 | Error | |
| InvalidAssociationObject | 2 | Error | |
| InvalidTcpObject | 3 | Error | |

6.8.1.34 ST_TcSocketsReceiverEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der Socket-Ereignisklasse für den Empfang der Daten.

Namensraum: [TC_EVENTS](#) [▶ 489].TcSocketsReceiverEventClass

Typ: Globale Variable

Bibliothek: Tc3_Sockets (Tc3_Sockets.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidPrimitiveState | 1 | Error | |
| InvalidAssociationObject | 2 | Error | |
| AckTimeoutError | 3 | Error | |
| InvalidParameterValue | 4 | Error | |
| FatalError | 5 | Error | |
| RcvBufferFreeFailure | 6 | Error | |

6.8.1.35 ST_TcSocketsSenderEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der Socket-Ereignisklasse für den Versand der Daten.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcSocketsSenderEventClass

Typ: Globale Variable

Bibliothek: Tc3_Sockets (Tc3_Sockets.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| SenderListFailure | 1 | Error | |
| InvalidPrimitiveState | 2 | Error | |
| InvalidTransactionObject | 3 | Error | |

6.8.1.36 ST_TcSocketsTransactionEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der Socket-Ereignisklasse für die Verwaltung der Datentransaktionen.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcSocketsTransactionEventClass

Typ: Globale Variable

Bibliothek: Tc3_Sockets (Tc3_Sockets.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidAssociationObject | 1 | Error | |
| InvalidDataObject | 2 | Error | |
| InvalidPrimitiveState | 3 | Error | |

6.8.1.37 ST_TcSocketsWin32SrvEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der TF6310 TCP/IP Server Win32-Ereignisklasse.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcSocketsWin32SrvEventClass

Typ: Globale Variable

Bibliothek: Tc3_Sockets (Tc3_Sockets.compiled-library)

6.8.1.38 ST_TcSystemEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der TwinCAT System Ereignisklasse.

Namensraum: `TC_EVENTS` [[▶ 489](#)].TcSystemEventClass

Typ: Globale Variable

Bibliothek: Nicht erforderlich, TwinCAT System Resource ab v3.1.4022.14

Ereignisse

| Name | ID | | Severity | Beschreibung |
|-----------------------------|------|-----|----------|--|
| | Hex | Dez | | |
| InternalError | 0x1 | 1 | Error | Interner Fehler |
| NoRTime | 0x2 | 2 | Error | Keine Echtzeit |
| AllocationLockedMemoryError | 0x3 | 3 | Error | Zuweisung gesperrt-Speicher fehler |
| InsertMailboxError | 0x4 | 4 | Error | Postfach voll |
| WrongReceiveHMSG | 0x5 | 5 | Error | Falsches HMSG |
| TargetPortNotFound | 0x6 | 6 | Error | Ziel-Port nicht gefunden |
| TargetMachineNotFound | 0x7 | 7 | Error | Zielrechner nicht gefunden |
| UnknownCommandID | 0x8 | 8 | Error | Unbekannte Befehl-ID |
| BadTaskID | 0x9 | 9 | Error | Ungültige Task-ID |
| NoIO | 0xA | 10 | Error | Kein IO |
| UnknownAdsCommand | 0xB | 11 | Error | Unbekannter ADS-Befehl |
| Win32Error | 0xC | 12 | Error | Win32 Fehler |
| PortNotConnected | 0xD | 13 | Error | Port nicht angeschlossen |
| InvalidAdsLength | 0xE | 14 | Error | Ungültige ADS-Länge |
| InvalidAdsNetID | 0xF | 15 | Error | Ungültige AMS Net ID |
| LowInstallationLevel | 0x10 | 16 | Error | niedrige Installationsebene |
| NoDebugAvailable | 0x11 | 17 | Error | Kein Debugging verfügbar |
| PortDisabled | 0x12 | 18 | Error | Port deaktiviert |
| PortAlreadyConnected | 0x13 | 19 | Error | Port bereits verbunden |
| AdsSyncWin32Error | 0x14 | 20 | Error | ADS Sync Win32 Fehler |
| AdsSyncTimeout | 0x15 | 21 | Error | ADS Sync Timeout |
| AdsSyncAmsError | 0x16 | 22 | Error | ADS Sync AMS Fehler |
| AdsSyncNoIndexMap | 0x17 | 23 | Error | Keine Index-Map für ADS Sync vorhanden |
| InvalidAdsPort | 0x18 | 24 | Error | Ungültiger ADS-Port |
| NoMemory | 0x19 | 25 | Error | Kein Speicher |
| TCPSendError | 0x1A | 26 | Error | TCP Sendefehler |
| HostUnreachable | 0x1B | 27 | Error | Host nicht erreichbar |
| InvalidAMSFragment | 0x1C | 28 | Error | Ungültiges AMS Fragment |

6.8.1.39 ST_TcTpktAbortEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der TPKT-Ereignisklasse für den Verbindungsabbruch.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcTpktAbortEventClass

Typ: Globale Variable

Bibliothek: Tc3_Tpkt (Tc3_Tpkt.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidAssociationObject | 1 | Error | |
| InvalidSocketObject | 2 | Error | |

6.8.1.40 ST_TcTpktAssociationEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der TPKT-Ereignisklasse für die Verwaltung einer Verbindungs-Assoziation.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcTpktAssociationEventClass

Typ: Globale Variable

Bibliothek: Tc3_Tpkt (Tc3_Tpkt.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|-------------------------|----|----------|--------------|
| InvalidEnvironmentState | 1 | Error | |

6.8.1.41 ST_TcTpktPduEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der TPKT-Ereignisklasse für die Verarbeitung der PDUs.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcTpktPduEventClass

Typ: Globale Variable

Bibliothek: Tc3_Tpkt (Tc3_Tpkt.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|---------------------------|----|----------|--------------|
| InvalidInputStreamObject | 1 | Error | |
| InvalidOutputStreamObject | 2 | Error | |
| InvalidHeaderSize | 3 | Error | |
| InvalidHeaderVersion | 4 | Error | |
| InvalidHeaderReserved | 5 | Error | |
| InvalidHeaderLength | 6 | Error | |

6.8.1.42 ST_TcTpktReceiverEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der TPKT-Ereignisklasse für den Empfang der Daten.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcTpktReceiverEventClass

Typ: Globale Variable

Bibliothek: Tc3_Tpkt (Tc3_Tpkt.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidPrimitiveState | 1 | Error | |
| InvalidAssociationObject | 2 | Error | |
| InvalidSocketObject | 3 | Error | |
| DataRspTimeoutError | 4 | Error | |
| DataResponseAbort | 5 | Error | |

6.8.1.43 ST_TcTpktSenderEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der TPKT-Ereignisklasse für den Versand der Daten.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcTpktSenderEventClass

Typ: Globale Variable

Bibliothek: Tc3_Tpkt (Tc3_Tpkt.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidAssociationObject | 1 | Error | |
| InvalidSocketObject | 2 | Error | |
| InvalidDataObject | 3 | Error | |

6.8.1.44 ST_TcTpktTransactionEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der TPKT-Ereignisklasse für die Verwaltung der Datentransaktionen.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcTpktTransactionEventClass

Typ: Globale Variable

Bibliothek: Tc3_Tpkt (Tc3_Tpkt.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidPrimitiveState | 1 | Error | |
| InvalidAssociationObject | 2 | Error | |
| InvalidSocketObject | 3 | Error | |

6.8.1.45 ST_TcUlosiAbortEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der ULOSI-Ereignisklasse für den Verbindungsabbruch.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcUlosiAbortEventClass

Typ: Globale Variable

Bibliothek: Tc3_Ulosi (Tc3_Ulosi.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidAssociationObject | 1 | Error | |
| InvalidRfc1006Object | 2 | Error | |

6.8.1.46 ST_TcUlosiAssociationEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der ULOSI-Ereignisklasse für die Verwaltung einer Verbindungs-Assoziation.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcUlosiAssociationEventClass

Typ: Globale Variable

Bibliothek: Tc3_Ulosi (Tc3_Ulosi.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|----------------------|----|----------|--------------|
| StateTransitionError | 1 | Error | |
| StateValidationError | 2 | Error | |

6.8.1.47 ST_TcUlosiClientEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der ULOSI-Ereignisklasse für die Verwaltung einer Client-Instanz.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcUlosiClientEventClass

Typ: Globale Variable

Bibliothek: Tc3_Ulosi (Tc3_Ulosi.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidAssociationObject | 1 | Error | |

6.8.1.48 ST_TcUlosiIncomingConnectEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der ULOSI-Ereignisklasse für den eingehenden Verbindungsaufbau.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcUlosiIncomingConnectEventClass

Typ: Globale Variable

Bibliothek: Tc3_Ulosi (Tc3_Ulosi.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidPrimitiveState | 1 | Error | |
| InvalidAssociationObject | 2 | Error | |
| InvalidSettingsObject | 3 | Error | |
| InvalidRfc1006Object | 4 | Error | |
| InvalidConnectionObject | 5 | Error | |
| InvalidServerObject | 6 | Error | |
| AARQTimeoutError | 7 | Error | |
| AARETimeoutError | 8 | Error | |

6.8.1.49 ST_TcUlosilIncomingDisconnectEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der ULOSI-Ereignisklasse für den eingehenden Verbindungsabbau.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcUlosilIncomingDisconnectEventClass

Typ: Globale Variable

Bibliothek: Tc3_Ulosi (Tc3_Ulosi.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidPrimitiveState | 1 | Error | |
| InvalidAssociationObject | 2 | Error | |
| InvalidSettingsObject | 3 | Error | |
| InvalidRfc1006Object | 4 | Error | |
| RLRETimeoutError | 5 | Error | |
| ConfirmationTimeoutError | 6 | Error | |

6.8.1.50 ST_TcUlosiOutgoingConnectEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der ULOSI-Ereignisklasse für den ausgehenden Verbindungsaufbau.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcUlosiOutgoingConnectEventClass

Typ: Globale Variable

Bibliothek: Tc3_Ulosi (Tc3_Ulosi.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidPrimitiveState | 1 | Error | |
| InvalidAssociationObject | 2 | Error | |
| InvalidSettingsObject | 3 | Error | |
| InvalidRfc1006Object | 4 | Error | |
| InvalidClientObject | 5 | Error | |
| AARETimeoutError | 6 | Error | |

6.8.1.51 ST_TcUlosiOutgoingDisconnectEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der ULOSI-Ereignisklasse für den ausgehenden Verbindungsabbau.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcUlosiOutgoingDisconnectEventClass

Typ: Globale Variable

Bibliothek: Tc3_Ulosi (Tc3_Ulosi.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidPrimitiveState | 1 | Error | |
| InvalidAssociationObject | 2 | Error | |
| InvalidRfc1006Object | 3 | Error | |
| RLRETimeoutError | 4 | Error | |

6.8.1.52 ST_TcUlosiReceiverEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der ULOSI-Ereignisklasse für den Empfang der Daten.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcUlosiReceiverEventClass

Typ: Globale Variable

Bibliothek: Tc3_Ulosi (Tc3_Ulosi.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidAssociationObject | 1 | Error | |
| InvalidSettingsObject | 2 | Error | |
| InvalidRfc1006Object | 3 | Error | |
| InvalidPrimitiveState | 4 | Error | |
| ResponseTimeoutError | 5 | Error | |
| DecoderFailure | 6 | Error | |

6.8.1.53 ST_TcUlosiSenderEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der ULOSI-Ereignisklasse für den Versand der Daten.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcUlosiSenderEventClass

Typ: Globale Variable

Bibliothek: Tc3_Ulosi (Tc3_Ulosi.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|-----------------------|----|----------|--------------|
| InvalidPrimitiveState | 1 | Error | |

6.8.1.54 ST_TcUlosiTransactionEventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der ULOSI-Ereignisklasse für die Verwaltung der Datentransaktionen.

Namensraum: [TC_EVENTS \[▶ 489\]](#).TcUlosiTransactionEventClass

Typ: Globale Variable

Bibliothek: Tc3_Ulosi (Tc3_Ulosi.compiled-library)

Ereignisse

| Name | ID | Severity | Beschreibung |
|--------------------------|----|----------|--------------|
| InvalidAssociationObject | 1 | Error | |
| InvalidRfc1006Object | 2 | Error | |
| EncoderFailure | 3 | Error | |

6.8.1.55 ST_Win32EventClass

Die folgende Tabelle enthält eine Liste der Ereignisse der Win32-Ereignisklasse.

- Ereignisse (IDs [0..99](#) [[▶ 520](#)]),
- Ereignisse (IDs [100..199](#) [[▶ 522](#)]),
- Ereignisse (IDs [200..1000](#) [[▶ 525](#)]),
- Ereignisse (IDs [1001..1099](#) [[▶ 527](#)]),
- Ereignisse (IDs [1100..1199](#) [[▶ 530](#)]),
- Ereignisse (IDs [1200..1399](#) [[▶ 533](#)]),
- Ereignisse (IDs [1400..1599](#) [[▶ 539](#)]),
- Ereignisse (IDs [1600..1799](#) [[▶ 541](#)]),
- Ereignisse (IDs [1800..1999](#) [[▶ 546](#)]),
- Ereignisse (IDs [2000..2999](#) [[▶ 548](#)]),
- Ereignisse (IDs [3000..4999](#) [[▶ 549](#)]),
- Ereignisse (IDs [5000..5999](#) [[▶ 552](#)]),
- Ereignisse (IDs [6000..7999](#) [[▶ 556](#)]),
- Ereignisse (IDs [8000..8499](#) [[▶ 558](#)]),
- Ereignisse (IDs [8500..9000](#) [[▶ 568](#)]),
- Ereignisse (IDs [9001..10003](#) [[▶ 573](#)]),
- Ereignisse (IDs [10004..11999](#) [[▶ 575](#)]),
- Ereignisse (IDs [12000..13884](#) [[▶ 578](#)])

Namensraum: [TC_EVENTS](#) [[▶ 489](#)].Win32EventClass

Typ: Globale Variable

Bibliothek: Nicht erforderlich, TwinCAT System Resource ab v3.1.4022.14

| Fehler | | | Beschreibung |
|---------|-------------|-------------------------------|--|
| Dezimal | Hexadezimal | Name | |
| 0 | 0x00000000 | ERROR_SUCCESS | Der Vorgang wurde erfolgreich abgeschlossen. |
| 1 | 0x00000001 | ERROR_INVALID_FUNCTION | Unzulässige Funktion. |
| 2 | 0x00000002 | ERROR_FILE_NOT_FOUND | Das System kann die angegebene Datei nicht finden. |
| 3 | 0x00000003 | ERROR_PATH_NOT_FOUND | Das System kann den angegebenen Pfad nicht finden. |
| 4 | 0x00000004 | ERROR_TOO_MANY_OPEN_FILES | Das System kann die Datei nicht öffnen. |
| 5 | 0x00000005 | ERROR_ACCESS_DENIED | Zugriff wird verweigert. |
| 6 | 0x00000006 | ERROR_INVALID_HANDLE | Das Handle ist ungültig. |
| 7 | 0x00000007 | ERROR_ARENA_TRASHED | Die Speicher-Kontrollblöcke wurden zerstört. |
| 8 | 0x00000008 | ERROR_NOT_ENOUGH_MEMORY | Es ist nicht genügend Speicher verfügbar, um diesen Befehl zu verarbeiten. |
| 9 | 0x00000009 | ERROR_INVALID_BLOCK | Die Speicher-Blockadresse ist ungültig. |
| 10 | 0x0000000A | ERROR_BAD_ENVIRONMENT | Die Umgebung ist nicht korrekt. |
| 11 | 0x0000000B | ERROR_BAD_FORMAT | Es wurde versucht, ein Programm mit einem falschen Format zu laden. |
| 12 | 0x0000000C | ERROR_INVALID_ACCESS | Der Zugangscode ist ungültig. |
| 13 | 0x0000000D | ERROR_INVALID_DATA | Die Daten sind ungültig. |
| 14 | 0x0000000E | ERROR_OUTOFMEMORY | Nicht genügend Speicherplatz ist zum Abschließen dieses Vorgangs verfügbar. |
| 15 | 0x0000000F | ERROR_INVALID_DRIVE | Das System kann das angegebene Laufwerk nicht finden. |
| 16 | 0x00000010 | ERROR_CURRENT_DIRECTORY | Das Verzeichnis kann nicht entfernt werden. |
| 17 | 0x00000011 | ERROR_NOT_SAME_DEVICE | Das System kann nicht die Datei in ein anderes Laufwerk verschieben. |
| 18 | 0x00000012 | ERROR_NO_MORE_FILES | Es gibt keine weiteren Dateien. |
| 19 | 0x00000013 | ERROR_WRITE_PROTECT | Das Medium ist schreibgeschützt. |
| 20 | 0x00000014 | ERROR_BAD_UNIT | Das System kann das angegebene Gerät nicht finden. |
| 21 | 0x00000015 | ERROR_NOT_READY | Das Gerät ist nicht bereit. |
| 22 | 0x00000016 | ERROR_BAD_COMMAND | Das Gerät erkennt den Befehl nicht. |
| 23 | 0x00000017 | ERROR_CRC | Datenfehler (CRC-Prüfung). |
| 24 | 0x00000018 | ERROR_BAD_LENGTH | Das Programm hat einen Befehl ausgegeben, aber die Befehlslänge ist falsch. |
| 25 | 0x00000019 | ERROR_SEEK | Das Laufwerk kann nicht suchen Sie einen bestimmten Bereich oder auf der Festplatte zu verfolgen. |
| 26 | 0x0000001A | ERROR_NOT_DOS_DISK | Die angegebene Festplatte oder Diskette kann nicht zugegriffen werden. |
| 27 | 0x0000001B | ERROR_SECTOR_NOT_FOUND | Das Laufwerk kann den angeforderten Sektor nicht finden. |
| 28 | 0x0000001C | ERROR_OUT_OF_PAPER | Der Drucker hat kein Papier. |
| 29 | 0x0000001D | ERROR_WRITE_FAULT | Das System kann nicht auf das angegebene Gerät schreiben. |
| 30 | 0x0000001E | ERROR_READ_FAULT | Das System kann nicht vom angegebenen Gerät lesen. |
| 31 | 0x0000001F | ERROR_GEN_FAILURE | Ein an das System angeschlossenes Gerät funktioniert nicht. |
| 32 | 0x00000020 | ERROR_SHARING_VIOLATION | Der Prozess kann nicht die Datei zugreifen, da es von einem anderen Prozess verwendet wird. |
| 33 | 0x00000021 | ERROR_LOCK_VIOLATION | Der Prozess kann nicht die Datei zugreifen, da ein anderer Prozess einen Teil der Datei gesperrt hat. |
| 34 | 0x00000022 | ERROR_WRONG_DISK | Die falsche Diskette befindet sich im Laufwerk. Fügen Sie %2 (Seriennummer des Datenträgers: %3) in das Laufwerk %1 ein. |
| 36 | 0x00000024 | ERROR_SHARING_BUFFER_EXCEEDED | Zu viele Dateien, die für die gemeinsame Nutzung geöffnet. |
| 38 | 0x00000026 | ERROR_HANDLE_EOF | Das Ende der Datei wurde erreicht. |
| 39 | 0x00000027 | ERROR_HANDLE_DISK_FULL | Der Datenträger ist voll. |
| 50 | 0x00000032 | ERROR_NOT_SUPPORTED | Die Anforderung wird nicht unterstützt. |
| 51 | 0x00000033 | ERROR_REM_NOT_LIST | Der Remotecomputer ist nicht verfügbar. |

| Fehler | | | Beschreibung |
|---------|-------------|-----------------------------|---|
| Dezimal | Hexadezimal | Name | |
| 52 | 0x00000034 | ERROR_DUP_NAME | Ein doppelter Name ist im Netzwerk vorhanden. |
| 53 | 0x00000035 | ERROR_BAD_NETPATH | Der Netzwerkpfad wurde nicht gefunden. |
| 54 | 0x00000036 | ERROR_NETWORK_BUSY | Das Netzwerk ist ausgelastet. |
| 55 | 0x00000037 | ERROR_DEV_NOT_EXIST | Die angegebene Netzwerkressource oder das Gerät ist nicht mehr verfügbar. |
| 56 | 0x00000038 | ERROR_TOO_MANY_CMDS | Das Netzwerk-BIOS-Befehl-Limit wurde erreicht. |
| 57 | 0x00000039 | ERROR_ADAP_HDW_ERR | Ein Netzwerk-Adapter-Hardware-Fehler aufgetreten. |
| 58 | 0x0000003A | ERROR_BAD_NET_RESP | Der angegebene Server kann nicht den angeforderten Vorgang ausführen. |
| 59 | 0x0000003B | ERROR_UNEXP_NET_ERR | Ein unerwarteter Netzwerkfehler ist aufgetreten. |
| 60 | 0x0000003C | ERROR_BAD_REM_ADAP | Der Remoteadapter ist nicht kompatibel. |
| 61 | 0x0000003D | ERROR_PRINTQ_FULL | Die Druckerwarteschlange ist voll. |
| 62 | 0x0000003E | ERROR_NO_SPOOL_SPACE | Speicherplatz zum Speichern der Datei, die darauf warten, gedruckt werden, ist nicht auf dem Server verfügbar. |
| 63 | 0x0000003F | ERROR_PRINT_CANCELLED | Ihre zum Drucken, wartende Datei wurde gelöscht. |
| 64 | 0x00000040 | ERROR_NETNAME_DELETED | Der angegebene Netzwerkname ist nicht mehr verfügbar. |
| 65 | 0x00000041 | ERROR_NETWORK_ACCESS_DENIED | Netzwerkzugriff wurde verweigert. |
| 66 | 0x00000042 | ERROR_BAD_DEV_TYPE | Der Netzwerk-Ressource-Typ ist nicht korrekt. |
| 67 | 0x00000043 | ERROR_BAD_NET_NAME | Der Netzwerkname kann nicht gefunden werden. |
| 68 | 0x00000044 | ERROR_TOO_MANY_NAMES | Die Namensbegrenzung für den lokalen Computer Netzwerkarte wurde überschritten. |
| 69 | 0x00000045 | ERROR_TOO_MANY_SESS | Die Netzwerk-BIOS-Sitzungsbegrenzung wurde überschritten. |
| 70 | 0x00000046 | ERROR_SHARING_PAUSED | Der Remoteserver wurde angehalten wurde oder wird gerade gestartet wird. |
| 71 | 0x00000047 | ERROR_REQ_NOT_ACCEP | Keine weitere Verbindungen können zu dieser Zeit mit diesem Remotecomputer erfolgen, denn es gibt bereits so viele Verbindungen wie der Computer annehmen kann. |
| 72 | 0x00000048 | ERROR_REDIR_PAUSED | Der angegebene Drucker oder das angegebene Datenträgergerät wurde angehalten. |
| 80 | 0x00000050 | ERROR_FILE_EXISTS | Die Datei ist vorhanden. |
| 82 | 0x00000052 | ERROR_CANNOT_MAKE | Das Verzeichnis oder die Datei kann nicht erstellt werden. |
| 83 | 0x00000053 | ERROR_FAIL_I24 | Fehler auf INT 24. |
| 84 | 0x00000054 | ERROR_OUT_OF_STRUCTURES | Speicher zur Verarbeitung dieser Anforderung ist nicht verfügbar. |
| 85 | 0x00000055 | ERROR_ALREADY_ASSIGNED | Der lokale Gerätenamen wird bereits verwendet. |
| 86 | 0x00000056 | ERROR_INVALID_PASSWORD | Das angegebene Netzwerkennwort ist falsch. |
| 87 | 0x00000057 | ERROR_INVALID_PARAMETER | Die Parameter ist falsch. |
| 88 | 0x00000058 | ERROR_NET_WRITE_FAULT | Ein Schreibfehler ist aufgetreten im Netzwerk. |
| 89 | 0x00000059 | ERROR_NO_PROC_SLOTS | Das System kann zur Zeit keinen weiteren Prozess starten. |

| Fehler | | | Beschreibung |
|---------|--------------|---------------------------------|--|
| Dezimal | Hexadezimal | Name | |
| 100 | 0x00000064 | ERROR_TOO_MANY_SEMAPHORES | Ein weiteres System Semaphore kann nicht erstellt werden. |
| 101 | 0 x 00000065 | ERROR_EXCL_SEM_ALREADY_OWNED | Die exklusive Semaphore gehört einem anderen Prozess. |
| 102 | 0x00000066 | ERROR_SEM_IS_SET | Das Semaphor ist festgelegt und kann nicht geschlossen werden. |
| 103 | 0x00000067 | ERROR_TOO_MANY_SEM_REQUESTS | Die Semaphore kann nicht erneut festgelegt werden. |
| 104 | 0x00000068 | ERROR_INVALID_AT_INTERRUPT_TIME | Keine anfordern exklusive Semaphore Interrupt Zeitpunkt. |
| 105 | 0x00000069 | ERROR_SEM_OWNER_DIED | Das frühere Eigentum an dieser Semaphore ist beendet. |
| 106 | 0x0000006A | ERROR_SEM_USER_LIMIT | Legen Sie die Diskette für Laufwerk %1. |
| 107 | 0x0000006B | ERROR_DISK_CHANGE | Das Programm beendet, da eine Alternative Diskette nicht eingelegt wurde. |
| 108 | 0x0000006C | ERROR_DRIVE_LOCKED | Der Datenträger wird verwendet oder von einem anderen Prozess gesperrt. |
| 109 | 0x0000006D | ERROR_BROKEN_PIPE | Die Pipe wurde beendet. |
| 110 | 0x0000006E | ERROR_OPEN_FAILED | Das System kann das Gerät oder die angegebene Datei nicht öffnen. |
| 111 | 0x0000006F | ERROR_BUFFER_OVERFLOW | Der Dateiname ist zu lang. |
| 112 | 0x00000070 | ERROR_DISK_FULL | Es ist nicht genügend Speicherplatz auf dem Datenträger. |
| 113 | 0x00000071 | ERROR_NO_MORE_SEARCH_HANDLES | Nicht mehr interne Datei-IDs zur Verfügung. |
| 114 | 0x00000072 | ERROR_INVALID_TARGET_HANDLE | Der interne Dateibezogener Ziel ist falsch. |
| 117 | 0x00000075 | ERROR_INVALID_CATEGORY | Der vom Anwendungsprogramm IOCTL-Aufruf ist nicht korrekt. |
| 118 | 0x00000076 | ERROR_INVALID_VERIFY_SWITCH | Der Wert des Parameters überprüfen-on-Write Switch ist nicht korrekt. |
| 119 | 0x00000077 | ERROR_BAD_DRIVER_LEVEL | Das System unterstützt den angeforderten Befehl nicht. |
| 120 | 0x00000078 | ERROR_CALL_NOT_IMPLEMENTED | Diese Funktion ist auf diesem System nicht unterstützt. |
| 121 | 0x00000079 | ERROR_SEM_TIMEOUT | Das Zeitlimit für die Semaphore ist abgelaufen. |
| 122 | 0x0000007A | ERROR_INSUFFICIENT_BUFFER | An einen Systemaufruf übergebene Datenbereich ist zu klein. |
| 123 | 0x0000007B | ERROR_INVALID_NAME | Der Dateiname, Verzeichnisname oder Datenträgerbezeichnung ist falsch. |
| 124 | 0x0000007C | ERROR_INVALID_LEVEL | Die Systemaufrufebene ist nicht korrekt. |
| 125 | 0x0000007D | ERROR_NO_VOLUME_LABEL | Der Datenträger hat keine Datenträgerbezeichnung. |
| 126 | 0x0000007E | ERROR_MOD_NOT_FOUND | Das angegebene Modul konnte nicht gefunden werden. |
| 127 | 0x0000007F | ERROR_PROC_NOT_FOUND | Die angegebene Prozedur konnte nicht gefunden werden. |
| 128 | 0x00000080 | ERROR_WAIT_NO_CHILDREN | Es gibt keine untergeordneten Prozesse warten. |
| 129 | 0x00000081 | ERROR_CHILD_NOT_COMPLETE | Die Anwendung %1 kann nicht im Win32-Modus ausgeführt werden. |
| 130 | 0x00000082 | ERROR_DIRECT_ACCESS_HANDLE | Versuch, ein Dateihandle einer offenen Datenträgerpartition für einen anderen Vorgang als raw Disk i/o zu verwenden. |
| 131 | 0x00000083 | ERROR_NEGATIVE_SEEK | Wurde versucht, den Dateizeiger vor den Anfang der Datei zu bewegen. |
| 132 | 0x00000084 | ERROR_SEEK_ON_DEVICE | Der Dateizeiger kann auf dem angegebenen Gerät oder Datei festgelegt werden. |
| 133 | 0x00000085 | ERROR_IS_JOIN_TARGET | Ein Join- oder SUBST-Befehl kann nicht für ein Laufwerk verwendet werden, die zuvor verbundene Laufwerke enthält. |
| 134 | 0x00000086 | ERROR_IS_JOINED | Wurde versucht, einen Join- oder SUBST-Befehl auf einem Laufwerk zu verwenden, die bereits verbunden worden ist. |
| 135 | 0x00000087 | ERROR_IS_SUBSTED | Es wurde versucht, einen Join- oder SUBST-Befehl auf einem Laufwerk zu verwenden, die bereits ersetzt worden ist. |

| Fehler | | | Beschreibung |
|---------|-------------|----------------------------------|---|
| Dezimal | Hexadezimal | Name | |
| 136 | 0x00000088 | ERROR_NOT_JOINED | Das System versucht, die Verknüpfung eines Laufwerks zu löschen, die nicht Mitglied ist. |
| 137 | 0x00000089 | ERROR_NOT_SUBSTED | Das System versucht, die Substitution eines Laufwerks zu löschen, die nicht ersetzt wird. |
| 138 | 0x0000008A | ERROR_JOIN_TO_JOIN | Das System versucht, ein Laufwerk zu einem Verzeichnis auf einem verknüpften Laufwerk anzuschließen. |
| 139 | 0x0000008B | ERROR_SUBST_TO_SUBST | Das System versucht, ein Laufwerk zu einem Verzeichnis auf einem subst Laufwerk zu ersetzen. |
| 140 | 0x0000008C | ERROR_JOIN_TO_SUBST | Das System versucht, mit dem Auto in ein Verzeichnis auf einem subst Laufwerk anzuschließen. |
| 141 | 0x0000008D | ERROR_SUBST_TO_JOIN | Das System versucht, SUBST ein Laufwerk zu einem Verzeichnis auf einem verknüpften Laufwerk. |
| 142 | 0x0000008E | ERROR_BUSY_DRIVE | Das System kann nicht zu dieser Zeit ein Join- oder SUBST ausführen. |
| 143 | 0x0000008F | ERROR_SAME_DRIVE | Das System kann nicht join oder ersetzen Sie mit dem Auto, oder für ein Verzeichnis auf dem gleichen Laufwerk. |
| 144 | 0x00000090 | ERROR_DIR_NOT_ROOT | Das Verzeichnis ist kein Unterverzeichnis des Stammverzeichnisses. |
| 145 | 0x00000091 | ERROR_DIR_NOT_EMPTY | Das Verzeichnis ist nicht leer. |
| 146 | 0x00000092 | ERROR_IS_SUBST_PATH | Der angegebene Pfad wird in Ersatz verwendet. |
| 147 | 0x00000093 | ERROR_IS_JOIN_PATH | Nicht genügend Ressourcen verfügbar, um diesen Befehl zu verarbeiten. |
| 148 | 0x00000094 | ERROR_PATH_BUSY | Der angegebene Pfad kann zu diesem Zeitpunkt verwendet werden. |
| 149 | 0x00000095 | ERROR_IS_SUBST_TARGET | Es wurde versucht, beitreten oder für die ein Verzeichnis auf dem Laufwerk das Ziel des vorherigen Ersatz ist mit dem Auto zu ersetzen. |
| 150 | 0x00000096 | ERROR_SYSTEM_TRACE | System-Trace-Informationen wurde in der CONFIG nicht angegeben.SYS-Datei, oder die Ablaufverfolgung ist nicht erlaubt. |
| 151 | 0x00000097 | ERROR_INVALID_EVENT_COUNT | Die Anzahl der angegebenen Semaphore-Ereignisse für DosMuxSemWait ist nicht korrekt. |
| 152 | 0x00000098 | ERROR_TOO_MANY_MUXWAITERS | DosMuxSemWait wurde nicht ausgeführt; zu viele Semaphoren sind bereits festgelegt. |
| 153 | 0x00000099 | ERROR_INVALID_LIST_FORMAT | Die DosMuxSemWait-Liste ist nicht korrekt. |
| 154 | 0x0000009A | ERROR_LABEL_TOO_LONG | Die eingegebene Datenträgerbezeichnung übersteigt die Etikett-Zeichen-Grenze des Dateisystems Ziel. |
| 155 | 0x0000009B | ERROR_TOO_MANY_TCBS | Ein anderer Thread kann nicht erstellt werden. |
| 156 | 0x0000009C | ERROR_SIGNAL_REFUSED | Der Empfängerprozess hat das Signal abgelehnt. |
| 157 | 0x0000009D | ERROR_DISCARDED | Das Segment ist bereits verworfen und kann nicht gesperrt werden. |
| 158 | 0x0000009E | ERROR_NOT_LOCKED | Das Segment ist bereits freigeschaltet. |
| 159 | 0x0000009F | ERROR_BAD_THREADID_ADDR | Die Adresse für die Thread-ID ist nicht korrekt. |
| 160 | 0x000000A0 | ERROR_BAD_ARGUMENTS | Die an DosExecPgm übergebene Argumentzeichenfolge ist falsch. |
| 161 | 0x000000A1 | ERROR_BAD_PATHNAME | Der angegebene Pfad ist ungültig. |
| 162 | 0x000000A2 | ERROR_SIGNAL_PENDING | Ein Signal ist bereits anhängig. |
| 164 | 0x000000A4 | ERROR_MAX_THRDS_REACHED | Keine weitere Threads können im System erstellt werden. |
| 167 | 0x000000A7 | ERROR_LOCK_FAILED | Nicht in der Lage, einen Bereich einer Datei zu sperren. |
| 170 | 0x000000AA | ERROR_BUSY | Die angeforderte Ressource ist in Verwendung. |
| 173 | 0x000000AD | ERROR_CANCEL_VIOLATION | Eine Sperranforderung war nicht hervorragend für die mitgelieferten "Abbrechen"-Region. |
| 174 | 0x000000AE | ERROR_ATOMIC_LOCKS_NOT_SUPPORTED | Das Dateisystem unterstützt keine atomare Änderungen an der Sperrentyp. |
| 180 | 0x000000B4 | ERROR_INVALID_SEGMENT_NUMBER | Erkennung des Systems durch eine Segment-Zahl, die nicht korrekt war. |
| 182 | 0x000000B6 | ERROR_INVALID_ORDINAL | Das Betriebssystem kann nicht %1 ausgeführt. |

| Fehler | | | Beschreibung |
|---------|-------------|---------------------------------|--|
| Dezimal | Hexadezimal | Name | |
| 183 | 0x000000B7 | ERROR_ALREADY_EXISTS | Eine Datei kann nicht erstellt werden, wenn die Datei bereits vorhanden ist. |
| 186 | 0x000000BA | ERROR_INVALID_FLAG_NUMBER | Das Flag übergeben ist nicht korrekt. |
| 187 | 0x000000BB | ERROR_SEM_NOT_FOUND | Der Namen des angegebenen Semaphore wurde nicht gefunden. |
| 188 | 0x000000BC | ERROR_INVALID_STARTING_CODESEG | Das Betriebssystem kann nicht %1 ausgeführt. |
| 189 | 0x000000BD | ERROR_INVALID_STACKSEG | Das Betriebssystem kann nicht %1 ausgeführt. |
| 190 | 0x000000BE | ERROR_INVALID_MODULETYPE | Das Betriebssystem kann nicht %1 ausgeführt. |
| 191 | 0x000000BF | ERROR_INVALID_EXE_SIGNATURE | %1 Kann nicht im Win32-Modus ausgeführt werden. |
| 192 | 0x000000C0 | ERROR_EXE_MARKED_INVALID | Das Betriebssystem kann nicht %1 ausgeführt. |
| 193 | 0x000000C1 | ERROR_BAD_EXE_FORMAT | %1 ist keine zulässige Win32-Anwendung. |
| 194 | 0x000000C2 | ERROR_ITERATED_DATA_EXCEEDS_64k | Das Betriebssystem kann nicht %1 ausgeführt. |
| 195 | 0x000000C3 | ERROR_INVALID_MINALLOCSIZE | Das Betriebssystem kann nicht %1 ausgeführt. |
| 196 | 0x000000C4 | ERROR_DYNLINK_FROM_INVALID_RING | Das Betriebssystem kann nicht dieses Anwendungsprogramm ausgeführt. |
| 197 | 0x000000C5 | ERROR_IOPL_NOT_ENABLED | Das Betriebssystem ist momentan nicht konfiguriert um diese Anwendung auszuführen. |
| 198 | 0x000000C6 | ERROR_INVALID_SEGDPL | Das Betriebssystem kann nicht %1 ausgeführt. |
| 199 | 0x000000C7 | ERROR_AUTODATASEG_EXCEEDS_64k | Das Betriebssystem kann nicht dieses Anwendungsprogramm ausgeführt. |

| Fehler | | | Beschreibung |
|---------|--|---------------------------------|--|
| Dezimal | Hexadezimal | Name | |
| 200 | 0x000000C8 | ERROR_RING2SEG_MUST_BE_MOVABLE | Das Code-Segment kann nicht größer als oder gleich 64 KB sein. |
| 201 | 0x000000c9 DRIVER_VERIFIER_IOMANAGER_VIOLATION | ERROR_RELOC_CHAIN_XEEDS_SEGLIM | Das Betriebssystem kann nicht %1 ausgeführt. |
| 202 | 0x000000CA | ERROR_INFLOOP_IN_RELOC_CHAIN | Das Betriebssystem kann nicht %1 ausgeführt. |
| 203 | 0x000000CB | ERROR_ENVVAR_NOT_FOUND | Das System nicht die Umgebungsoption gefunden, die eingegeben wurde. |
| 205 | 0x000000CD | ERROR_NO_SIGNAL_SENT | Kein Prozess in der Unterstruktur Befehl hat einen Signalhandler. |
| 206 | 0x000000CE | ERROR_FILENAME_EXCED_RANGE | Der Dateiname oder die Erweiterung ist zu lang. |
| 207 | 0x000000CF | ERROR_RING2_STACK_IN_USE | Der Ring 2-Stapel wird verwendet. |
| 208 | 0x000000D0 | ERROR_META_EXPANSION_TOO_LONG | Die globale Dateinamenszeichen * oder?, sind falsch eingegeben oder es wurden zu viele globale Dateinamenszeichen angegeben. |
| 209 | 0x000000D1 | ERROR_INVALID_SIGNAL_NUMBER | Das Signal, das gesendet wird, ist nicht korrekt. |
| 210 | 0x000000D2 | ERROR_THREAD_1_INACTIVE | Der Signalhandler kann nicht festgelegt werden. |
| 212 | 0x000000D4 | ERROR_LOCKED | Das Segment ist gesperrt und kann nicht neu zugewiesen werden. |
| 214 | 0x000000D6 | ERROR_TOO_MANY_MODULES | An dieses Programm oder Dynamic Link Modul sind zu viele Dynamic Link Module angefügt. |
| 215 | 0x000000D7 | ERROR_NESTING_NOT_ALLOWED | Aufrufe von LoadModule können nicht verschachtelt werden. |
| 216 | 0x000000D8 | ERROR_EXE_MACHINE_TYPE_MISMATCH | Die Image-Datei "% 1" ist gültig, aber ist für einen Maschinentyp als dem aktuellen Computer. |
| 230 | 0x000000E6 | ERROR_BAD_PIPE | Der Rohr-Status ist ungültig. |
| 231 | 0x000000E7 | ERROR_PIPE_BUSY | Alle Pipe-Instanzen sind beschäftigt. |
| 232 | 0x000000E8 | ERROR_NO_DATA | Das Rohr ist geschlossen. |
| 233 | 0x000000E9 | ERROR_PIPE_NOT_CONNECTED | Kein Prozess ist am anderen Ende der Leitung. |
| 234 | 0x000000EA | ERROR_MORE_DATA | Weitere Daten sind verfügbar. |
| 240 | 0x000000F0 | ERROR_VC_DISCONNECTED | Die Sitzung wurde abgebrochen. |
| 254 | 0x000000FE | ERROR_INVALID_EA_NAME | Das angegebene erweiterte Attribut-Name ist ungültig. |
| 255 | 0x000000FF | ERROR_EA_LIST_INCONSISTENT | Die erweiterten Attribute sind inkonsistent. |
| 258 | 0x00000102 | WAIT_TIMEOUT | Der Wartevorgang Zeitüberschreitung. |
| 259 | 0x00000103 | ERROR_NO_MORE_ITEMS | Keine weiteren Daten ist verfügbar. |
| 266 | 0x0000010A | ERROR_CANNOT_COPY | Die Kopierfunktionen können nicht verwendet werden. |
| 267 | 0x0000010B | ERROR_DIRECTORY | Der Verzeichnisname ist ungültig. |
| 275 | 0x00000113 | ERROR_EAS_DIDNT_FIT | Die erweiterten Attribute passten nicht in den Puffer. |
| 276 | 0x00000114 | ERROR_EA_FILE_CORRUPT | Die erweiterte-Attribute-Datei auf dem System bereitgestellte Datei ist beschädigt. |
| 277 | 0x00000115 | ERROR_EA_TABLE_FULL | Die erweiterte-Attribute-Datei ist voll. |
| 278 | 0x00000116 | ERROR_INVALID_EA_HANDLE | Das angegebene erweiterte Attribut Handle ist ungültig. |
| 282 | 0x0000011A | ERROR_EAS_NOT_SUPPORTED | Das bereitgestellte Dateisystem unterstützt keine erweiterten Attribute. |
| 288 | 0x00000120 | ERROR_NOT_OWNER | Versuchen Sie, nicht im Besitz des Aufrufers Mutex freizugeben. |
| 298 | 0x0000012A | ERROR_TOO_MANY_POSTS | Zu viele Beiträge wurden auf einen Semaphor. |
| 299 | 0x0000012B | ERROR_PARTIAL_COPY | Nur ein Teil der ReadProcessMemory- oder WriteProcessMemory-Anforderung wurde abgeschlossen. |
| 300 | 0x0000012C | ERROR_OPLOCK_NOT_GRANTED | Die Oplock-Anforderung wurde verweigert. |
| 301 | 0x0000012D | ERROR_INVALID_OPLOCK_PROTOCOL | Eine ungültige Oplock-Bestätigung wurde vom System empfangen. |
| 302 | 0x0000012E | ERROR_DISK_TOO_FRAGMENTED | Das Volume ist zu fragmentiert, um diesen Vorgang abzuschließen. |
| 303 | 0x0000012F | ERROR_DELETE_PENDING | Die Datei kann nicht geöffnet werden, da es wird gerade gelöscht wird. |

| Fehler | | | Beschreibung |
|---------|-------------|---------------------------|--|
| Dezimal | Hexadezimal | Name | |
| 317 | 0x0000013D | ERROR_MR_MID_NOT_FOUND | Das System kann nicht finden Meldungstext für die Meldungsnummer 0 x %1 in der Meldungsdatei für %2. |
| 487 | 0x000001E7 | ERROR_INVALID_ADDRESS | Versuch, ungültige Adresse zuzugreifen. |
| 534 | 0x00000216 | ERROR_ARITHMETIC_OVERFLOW | Arithmetisches Ergebnis übertraf 32 Bits. |
| 535 | 0x00000217 | ERROR_PIPE_CONNECTED | Es ist ein Prozess am anderen Ende der Pipe. |
| 536 | 0x00000218 | ERROR_PIPE_LISTENING | Ich warte auf ein Prozess das andere Ende der Pipe zu öffnen. |
| 994 | 0x000003E2 | ERROR_EA_ACCESS_DENIED | Zugriff auf die erweiterte-Attribute wurde verweigert. |
| 995 | 0x000003E3 | ERROR_OPERATION_ABORTED | Die i/o-Operation wurde wegen eines Threadendes oder einer Anwendungsanforderung abgebrochen. |
| 996 | 0x000003E4 | ERROR_IO_INCOMPLETE | Überlappendes I/O Ereignis befindet sich nicht in einem signalisierten Zustand. |
| 997 | 0x000003E5 | ERROR_IO_PENDING | Überlappenden i/o-Operation ist im Gange. |
| 998 | 0x000003E6 | ERROR_NOACCESS | Ungültiger Zugriff auf Speicherbereich. |
| 999 | 0x000003E7 | ERROR_SWAPERROR | Fehler beim Speicherseite. |

| Fehler | | | Beschreibung |
|---------|-------------|----------------------------------|--|
| Dezimal | Hexadezimal | Name | |
| 1001 | 0x000003E9 | ERROR_STACK_OVERFLOW | Rekursion zu tief; der Stapel ist übergelaufen. |
| 1002 | 0x000003EA | ERROR_INVALID_MESSAGE | Das Fenster kann nicht auf die gesendete Nachricht fungieren. |
| 1003 | 0x000003EB | ERROR_CAN_NOT_COMPLETE | Diese Funktion kann nicht abgeschlossen werden. |
| 1004 | 0x000003EC | ERROR_INVALID_FLAGS | Ungültige Flags. |
| 1005 | 0x000003ED | ERROR_UNRECOGNIZED_VOLUME | Das Volumen ist keine anerkannte Dateisystem enthalten. Bitte achten Sie darauf, dass alle erforderliche Datei-System-Treiber geladen sind und dass die Lautstärke nicht beschädigt ist. |
| 1006 | 0x000003EE | ERROR_FILE_INVALID | Das Volumen für eine Datei wurde extern geändert, so dass die geöffnete Datei nicht mehr gültig ist. |
| 1007 | 0x000003EF | ERROR_FULLSCREEN_MODE | Der angeforderte Vorgang kann nicht im Vollbild Modus ausgeführt werden. |
| 1008 | 0x000003F0 | ERROR_NO_TOKEN | Es wurde versucht, auf ein Token zu verweisen, die nicht vorhanden ist. |
| 1009 | 0x000003F1 | ERROR_BADDB | Die Konfiguration-Registry-Datenbank ist beschädigt. |
| 1010 | 0x000003F2 | ERROR_BADKEY | Der Schlüssel der Konfigurationsregistrierung ist ungültig. |
| 1011 | 0x000003F3 | ERROR_CANTOPEN | Der Schlüssel der Konfigurationsregistrierung konnte nicht geöffnet werden. |
| 1012 | 0x000003F4 | ERROR_CANTREAD | Der Schlüssel der Konfigurationsregistrierung konnte nicht gelesen werden. |
| 1013 | 0x000003F5 | ERROR_CANTWRITE | Der Schlüssel der Konfigurationsregistrierung konnte nicht geschrieben werden. |
| 1014 | 0x000003F6 | ERROR_REGISTRY_RECOVERED | Eine der Dateien in der Registrierungsdatenbank musste durch Verwendung von ein Protokoll oder eine Sicherungskopie wiederhergestellt werden. Die Wiederherstellung war erfolgreich. |
| 1015 | 0x000003F7 | ERROR_REGISTRY_CORRUPT | Die Registrierung ist beschädigt. Die Struktur einer der Dateien, die Registrierungsdaten enthält, ist beschädigt, oder Speicherabzug der Datei beschädigt ist oder die Datei konnte nicht wiederhergestellt werden, da die Sicherungskopie oder das Protokoll fehlte oder beschädigt. |
| 1016 | 0x000003F8 | ERROR_REGISTRY_IO_FAILED | Eine i/o-Operation durch die Registrierung ausgelöst, ist fehlgeschlagen. Die Registrierung konnte nicht lesen, oder ausschreiben, oder flush, eine der Dateien, die das Systemabbild der Registrierung enthalten. |
| 1017 | 0x000003F9 | ERROR_NOT_REGISTRY_FILE | Das System hat versucht, zu laden oder eine Datei in die Registrierung wiederherstellen, aber die angegebene Datei ist nicht in einem Registrierungs-Dateiformat. |
| 1018 | 0x000003FA | ERROR_KEY_DELETED | Unzulässiger Vorgang auf einen Registrierungsschlüssel, der zum Löschen markiert wurde versucht. |
| 1019 | 0x000003FB | ERROR_NO_LOG_SPACE | Das System konnte den benötigten Speicherplatz in einem Registrierungsprotokoll nicht reservieren. |
| 1020 | 0x000003FC | ERROR_KEY_HAS_CHILDREN | Eine symbolische Verknüpfung kann nicht in einem Registrierungsschlüssel erstellt werden, die bereits Unterschlüssel oder Werte hat. |
| 1021 | 0x000003FD | ERROR_CHILD_MUST_BE_VOLATILE | Einen dauerhaften Unterschlüssel unter einem temporären übergeordneten Schlüssel kann nicht erstellt werden. |
| 1022 | 0x000003FE | ERROR_NOTIFY_ENUM_DIR | Eine Anforderung einer Änderungsnachricht ist abgeschlossen, und die Informationen wird nicht im Puffer des Aufrufers zurückgegeben. Der Aufrufer muss jetzt die Dateien um die Änderungen zu finden auflisten. |
| 1051 | 0x0000041B | ERROR_DEPENDENT_SERVICES_RUNNING | Ein Stopzeichen wurde an einen Dienst gesendet, die andere Dienste abhängen. |
| 1052 | 0x0000041C | ERROR_INVALID_SERVICE_CONTROL | Das angeforderte Steuerelement ist nicht gültig für diesen Dienst. |

| Fehler | | | Beschreibung |
|---------|-------------|---|---|
| Dezimal | Hexadezimal | Name | |
| 1053 | 0x0000041D | ERROR_SERVICE_REQUEST_TIMEOUT | Der Dienst hat nicht rechtzeitig auf die Start- oder reagiert. |
| 1054 | 0x0000041E | ERROR_SERVICE_NO_THREAD | Ein Thread konnte nicht für den Dienst erstellt werden. |
| 1055 | 0x0000041F | ERROR_SERVICE_DATABASE_LOCKED | Die Dienstdatenbank ist gesperrt. |
| 1056 | 0x00000420 | ERROR_SERVICE_ALREADY_RUNNING | Bereits ist eine Instanz des Dienstes ausgeführt. |
| 1057 | 0x00000421 | ERROR_INVALID_SERVICE_ACCOUNT | Der Kontoname ist ungültig oder nicht vorhanden, oder das Kennwort ist ungültig für den Kontonamen angegeben. |
| 1058 | 0x00000422 | ERROR_SERVICE_DISABLED | Der Dienst kann nicht gestartet werden, weil es deaktiviert ist oder weil sie keine aktivierten Geräte zugeordnet hat. |
| 1059 | 0x00000423 | ERROR_CIRCULAR_DEPENDENCY | Kreisförmige Dienstabhängigkeit wurde angegeben. |
| 1060 | 0x00000424 | ERROR_SERVICE_DOES_NOT_EXIST | Der angegebene Dienst ist nicht als installierter Dienst vorhanden. |
| 1061 | 0x00000425 | ERROR_SERVICE_CANNOT_ACCEPT_CTRL | Der Dienst kann nicht diesmal Steuerungsmeldungen annehmen. |
| 1062 | 0x00000426 | ERROR_SERVICE_NOT_ACTIVE | Der Dienst wurde nicht gestartet. |
| 1063 | 0x00000427 | ERROR_FAILED_SERVICE_CONTROLLER_CONNECT | Der Dienstprozess konnte keine Verbindung den Dienstcontroller herstellen. |
| 1064 | 0x00000428 | ERROR_EXCEPTION_IN_SERVICE | Eine Ausnahme ist in den Dienst beim Umgang mit der Anforderung. |
| 1065 | 0x00000429 | ERROR_DATABASE_DOES_NOT_EXIST | Die angegebene Datenbank existiert nicht. |
| 1066 | 0x0000042A | ERROR_SERVICE_SPECIFIC_ERROR | Der Dienst hat einen dienstspezifischen Fehlercode zurückgegeben. |
| 1067 | 0x0000042B | ERROR_PROCESS_ABORTED | Der Prozess wurde unerwartet beendet. |
| 1068 | 0x0000042C | ERROR_SERVICE_DEPENDENCY_FAIL | Der Abhängigkeitsdienst oder die Gruppe konnte nicht gestartet werden. |
| 1069 | 0x0000042D | ERROR_SERVICE_LOGON_FAILED | Der Dienst wurde aufgrund eines Anmeldefehlers nicht gestartet. |
| 1070 | 0x0000042E | ERROR_SERVICE_START_HANG | Nach dem Start hing der Dienst einen Start-ausstehende Zustand. |
| 1071 | 0x0000042F | ERROR_INVALID_SERVICE_LOCK | Die angegebene Sperre für die Datenbank ist ungültig. |
| 1072 | 0x00000430 | ERROR_SERVICE_MARKED_FOR_DELETE | Der angegebene Dienst wurde zum Löschen markiert. |
| 1073 | 0x00000431 | ERROR_SERVICE_EXISTS | Der angegebene Dienst ist bereits vorhanden. |
| 1074 | 0x00000432 | ERROR_ALREADY_RUNNING_LKG | Das System wird derzeit mit der letzten funktionierenden Konfiguration ausgeführt. |
| 1075 | 0x00000433 | ERROR_SERVICE_DEPENDENCY_DELETED | Der Abhängigkeitsdienst existiert nicht oder wurde zum Löschen markiert. |
| 1076 | 0x00000434 | ERROR_BOOT_ALREADY_ACCEPTED | Die aktuelle Startkonfiguration wurde bereits für den Einsatz als letzte als funktionierend Steuerelementsatz akzeptiert. |
| 1077 | 0x00000435 | ERROR_SERVICE_NEVER_STARTED | Keine Versuche zum Starten des Dienstes wurden seit dem letzten Start. |
| 1078 | 0x00000436 | ERROR_DUPLICATE_SERVICE_NAME | Der Name wird bereits verwendet als einen Dienstnamen oder einen Service-Anzeigenamen. |
| 1079 | 0x00000437 | ERROR_DIFFERENT_SERVICE_ACCOUNT | Für diesen Dienst angegebene Konto unterscheidet sich von dem für andere Dienste, die in demselben Prozess ausgeführt angegebene Konto. |
| 1080 | 0x00000438 | ERROR_CANNOT_DETECT_DRIVER_FAILURE | Fehleraktionen können nur für Win32-Dienste nicht für Treiber festgelegt werden. |
| 1081 | 0x00000439 | ERROR_CANNOT_DETECT_PROCESS_ABORT | Dieser Dienst wird in demselben Prozess wie der Dienststeuerungs-Manager ausgeführt. Daher kann nicht der Dienststeuerungs-Manager Maßnahmen zu ergreifen, falls dieser Dienst Prozess unerwartet beendet wird. |
| 1082 | 0x0000043A | ERROR_NO_RECOVERY_PROGRAM | Kein Wiederherstellungsprogramm wurde für diesen Dienst konfiguriert. |
| 1083 | 0x0000043B | ERROR_SERVICE_NOT_IN_EXE | Das ausführbare Programm, dass dieser Dienst für die Ausführung konfiguriert wird den Dienst nicht implementiert. |

| Fehler | | | Beschreibung |
|---------|-------------|----------------------------|---|
| Dezimal | Hexadezimal | Name | |
| 1084 | 0x0000043C | ERROR_NOT_SAFEBOOT_SERVICE | Dieser Dienst kann nicht im abgesicherten Modus gestartet werden. |

| Fehler | | | Beschreibung |
|---------|-------------|--------------------------------|---|
| Dezimal | Hexadezimal | Name | |
| 1100 | 0x0000044C | ERROR_END_OF_MEDIA | Das physische Ende des Bandes wurde erreicht. |
| 1101 | 0x0000044D | ERROR_FILEMARK_DETECTED | Ein Bandzugriff erreichte eine Dateimark. |
| 1102 | 0x0000044E | ERROR_BEGINNING_OF_MEDIA | Am Anfang des Bandes oder eine Partition wurde gefunden. |
| 1103 | 0x0000044F | ERROR_SETMARK_DETECTED | Ein Bandzugriff erreichte das Ende eines Satzes von Dateien. |
| 1104 | 0x00000450 | ERROR_NO_DATA_DETECTED | Keine weiteren Daten ist auf dem Band. |
| 1105 | 0x00000451 | ERROR_PARTITION_FAILURE | Band konnte nicht partitioniert werden. |
| 1106 | 0x00000452 | ERROR_INVALID_BLOCK_LENGTH | Beim Zugriff auf ein neues Band einer Partition über mehrere Datenträger ist die aktuelle Blockgröße falsch. |
| 1107 | 0x00000453 | ERROR_DEVICE_NOT_PARTITIONED | Band-Partitionsinformationen konnte nicht gefunden werden, beim Laden eines Bandes. |
| 1108 | 0x00000454 | ERROR_UNABLE_TO_LOCK_MEDIA | Konnte nicht gesperrt werden Auswerfen Mechanismus. |
| 1109 | 0x00000455 | ERROR_UNABLE_TO_UNLOAD_MEDIA | Nicht in der Lage, die Medien zu entladen. |
| 1110 | 0x00000456 | ERROR_MEDIA_CHANGED | Das Medium im Laufwerk möglicherweise geändert. |
| 1111 | 0x00000457 | ERROR_BUS_RESET | Der i/o-Bus wurde zurückgesetzt. |
| 1112 | 0x00000458 | ERROR_NO_MEDIA_IN_DRIVE | Kein Medium im Laufwerk. |
| 1113 | 0x00000459 | ERROR_NO_UNICODE_TRANSLATION | Keine Zuordnung für das Unicode-Zeichen existiert in der Ziel-Multi-Byte-Codepage. |
| 1114 | 0x0000045A | ERROR_DLL_INIT_FAILED | Eine dynamic Link Library (DLL) Initialisierungsroutine ist fehlgeschlagen. |
| 1115 | 0x0000045B | ERROR_SHUTDOWN_IN_PROGRESS | Ein Herunterfahren des Systems ist im Gange. |
| 1116 | 0x0000045C | ERROR_NO_SHUTDOWN_IN_PROGRESS | Nicht in der Lage, das System Herunterfahren abubrechen, da kein Herunterfahren ausgeführt wurde. |
| 1117 | 0x0000045D | ERROR_IO_DEVICE | Die Anforderung konnte nicht aufgrund eines i/o Device Fehler ausgeführt werden. |
| 1118 | 0x0000045E | ERROR_SERIAL_NO_DEVICE | Kein serielles Gerät wurde erfolgreich initialisiert. Der serielle Treiber wird entladen. |
| 1119 | 0x0000045F | ERROR_IRQ_BUSY | Nicht in der Lage, ein Gerät zu öffnen, die eine Interruptanforderung (IRQ) zusammen mit anderen Geräten teilnahm. Mindestens ein anderes Gerät, das diesen IRQ verwendet wurde bereits eröffnet. |
| 1120 | 0x00000460 | ERROR_MORE_WRITES | Eine serielle i/o-Operation wurde durch ein weiteres Schreiben an den seriellen Port abgeschlossen. (Die vervollständigt.
\\n(IOCTL_SERIAL_XOFF_COUNTER erreicht NULL.) |
| 1121 | 0x00000461 | ERROR_COUNTER_TIMEOUT | Eine serielle i/o-Operation abgeschlossen, da das Timeout ist abgelaufen. (Die vervollständigt.
\\n(IOCTL_SERIAL_XOFF_COUNTER hat nicht Null erreicht.) |
| 1122 | 0x00000462 | ERROR_FLOPPY_ID_MARK_NOT_FOUND | Auf der Diskette wurde keine ID-Adresse-Mark gefunden. |
| 1123 | 0x00000463 | ERROR_FLOPPY_WRONG_CYLINDER | Spuradresse Diskette Sektor-ID-Feld der Diskette Controller Track. |
| 1124 | 0x00000464 | ERROR_FLOPPY_UNKNOWN_ERROR | Der Diskettencontroller hat einen Fehler, der vom Diskettentreiber nicht erkannt wird gemeldet. |
| 1125 | 0x00000465 | ERROR_FLOPPY_BAD_REGISTERS | Der Diskettencontroller kehrte inkonsistente Ergebnisse in seinen Registern zurück. |
| 1126 | 0x00000466 | ERROR_DISK_RECALIBRATE_FAILED | Beim Zugriff auf die Festplatte, schlug ein neu kalibrieren, auch nach Wiederholungen. |
| 1127 | 0x00000467 | ERROR_DISK_OPERATION_FAILED | Bei Festplattenzugriff, schlug ein Festplattenvorgang auch nach Wiederholungen. |
| 1128 | 0x00000468 | ERROR_DISK_RESET_FAILED | Beim Zugriff auf die Festplatte, ein Festplatten-Controller-Reset war nötig, jedoch auch versagt. |
| 1129 | 0x00000469 | ERROR_EOM_OVERFLOW | Physischen Ende Bandende wurde erreicht. |
| 1130 | 0x0000046A | ERROR_NOT_ENOUGH_SERVER_MEMORY | Es ist nicht genügend Server-Speicherplatz verfügbar, um diesen Befehl zu verarbeiten. |
| 1131 | 0x0000046B | ERROR_POSSIBLE_DEADLOCK | Eine potenzielle Deadlock-Bedingung wurde erkannt. |

| Fehler | | | Beschreibung |
|---------|-------------|--------------------------------------|---|
| Dezimal | Hexadezimal | Name | |
| 1132 | 0x0000046C | ERROR_MAPPED_ALIGNMENT | Die Basisadresse oder der angegebene Dateioffset hat nicht die richtige Anordnung. |
| 1140 | 0x00000474 | ERROR_SET_POWER_STATE_VETOED | Ein Versuch, den Energiestatus zu ändern wurde durch eine andere Anwendung oder ein Treiber Veto. |
| 1141 | 0x00000475 | ERROR_SET_POWER_STATE_FAILED | Das System-BIOS konnte einen Versuch, den Energiestatus zu ändern. |
| 1142 | 0x00000476 | ERROR_TOO_MANY_LINKS | Wurde versucht, mehr Verknüpfungen auf eine Datei als das Dateisystem unterstützt erstellen. |
| 1150 | 0x0000047E | ERROR_OLD_WIN_VERSION | Das angegebene Programm benötigt eine neuere Version von Windows. |
| 1151 | 0x0000047F | ERROR_APP_WRONG_OS | Das angegebene Programm ist kein Windows- oder MS-DOS-Programm. |
| 1152 | 0x00000480 | ERROR_SINGLE_INSTANCE_APP | Mehr als eine Instanz der das angegebene Programm kann nicht gestartet werden. |
| 1153 | 0x00000481 | ERROR_RMODE_APP | Das angegebene Programm wurde für eine frühere Version von Windows geschrieben. |
| 1154 | 0x00000482 | ERROR_INVALID_DLL | Eine Bibliothekdatei, die zum Ausführen dieser Anwendung erforderlich ist beschädigt. |
| 1155 | 0x00000483 | ERROR_NO_ASSOCIATION | Keine Anwendung ist der angegebenen Datei für diesen Vorgang zugeordnet. |
| 1156 | 0x00000484 | ERROR_DDE_FAIL | Fehler beim Senden des Befehls an die Anwendung. |
| 1157 | 0x00000485 | ERROR_DLL_NOT_FOUND | Eine Bibliothekdatei, die zum Ausführen dieser Anwendung erforderlich kann nicht gefunden werden. |
| 1158 | 0x00000486 | ERROR_NO_MORE_USER_HANDLES | Der aktuelle Prozess hat alle Managerobjekte Griffe für Fenster-Manager-Objekte verwendet. |
| 1159 | 0x00000487 | ERROR_MESSAGE_SYNC_ONLY | Die Nachricht kann nur mit synchronen Vorgängen verwendet werden. |
| 1160 | 0x00000488 | ERROR_SOURCE_ELEMENT_EMPTY | Das angegebenen Quellelement verfügt über keine Medien. |
| 1161 | 0x00000489 | ERROR_DESTINATION_ELEMENT_FULL | Das angegebene Zielelement enthält bereits ein Medium. |
| 1162 | 0x0000048A | ERROR_ILLEGAL_ELEMENT_ADDRESS | Das angegebene Element ist nicht vorhanden. |
| 1163 | 0x0000048B | ERROR_MAGAZINE_NOT_PRESENT | Das angegebene Element ist Teil einer Zeitschrift, die nicht vorhanden ist. |
| 1164 | 0x0000048C | ERROR_DEVICE_REINITIALIZATION_NEEDED | Das angegebene Gerät muss wegen Hardwarefehler. |
| 1165 | 0x0000048D | ERROR_DEVICE_REQUIRES_CLEANING | Das Gerät hat angezeigt, dass die Reinigung erforderlich ist, bevor weitere Operationen durchgeführt werden. |
| 1166 | 0x0000048E | ERROR_DEVICE_DOOR_OPEN | Das Gerät hat angezeigt, dass die Tür offen ist. |
| 1167 | 0x0000048F | ERROR_DEVICE_NOT_CONNECTED | Das Gerät ist nicht angeschlossen. |
| 1168 | 0x00000490 | ERROR_NOT_FOUND | Element nicht gefunden. |
| 1169 | 0x00000491 | ERROR_NO_MATCH | Es gab keine Übereinstimmung für den angegebenen Schlüssel im Index. |
| 1170 | 0x00000492 | ERROR_SET_NOT_FOUND | Der angegebene Eigenschaftensatz ist nicht im Objekt vorhanden. |
| 1171 | 0x00000493 | ERROR_POINT_NOT_FOUND | Der Punkt an GetMouseMovePointsEx übergeben, ist nicht im Puffer. |
| 1172 | 0x00000494 | ERROR_NO_TRACKING_SERVICE | Der Überwachungsdienst (Arbeitsstation) wird nicht ausgeführt. |
| 1173 | 0x00000495 | ERROR_NO_VOLUME_ID | Die Volume-ID konnte nicht gefunden werden. |
| 1175 | 0x00000497 | ERROR_UNABLE_TO_REMOVE_REPLACED | Kann nicht zum Entfernen der Datei ersetzt werden. |
| 1176 | 0x00000498 | ERROR_UNABLE_TO_MOVE_REPLACEMENT | Können nicht die Ersatzdatei in die Datei ersetzt werden. Die Datei ersetzt werden, hat ihren ursprünglichen Namen behalten. |
| 1177 | 0x00000499 | ERROR_UNABLE_TO_MOVE_REPLACEMENT_2 | Können nicht die Ersatzdatei in die Datei ersetzt werden. Die Datei ersetzt werden wurde mithilfe der Name der Sicherungskopie umbenannt. |
| 1178 | 0x0000049A | ERROR_JOURNAL_DELETE_IN_PROGRESS | Das Band-Änderungsjournal wird gelöscht. |
| 1179 | 0x0000049B | ERROR_JOURNAL_NOT_ACTIVE | Das Band-Änderungsjournal ist nicht aktiv. |

| Fehler | | | Beschreibung |
|---------|-------------|-----------------------------|---|
| Dezimal | Hexadezimal | Name | |
| 1180 | 0x0000049C | ERROR_POTENTIAL_FILE_FOUND | Eine Datei wurde gefunden, aber es kann nicht sein, die richtige Datei. |
| 1181 | 0x0000049D | ERROR_JOURNAL_ENTRY_DELETED | Der Journaleintrag wurde aus dem Journal gelöscht. |

| Fehler | | | Beschreibung |
|---------|-------------|-------------------------------------|---|
| Dezimal | Hexadezimal | Name | |
| 1200 | 0x000004B0 | ERROR_BAD_DEVICE | Der angegebene Geräte-name ist ungültig. |
| 1201 | 0x000004B1 | ERROR_CONNECTION_UNAVAIL | Das Gerät ist momentan nicht verbunden, aber es ist eine Erinnerung Verbindung. |
| 1202 | 0x000004B2 | ERROR_DEVICE_ALREADY_REMEMBERED | Der lokale Geräte-name hat eine erinnerte Verbindung zu anderen Netzwerkressource. |
| 1203 | 0x000004B3 | ERROR_NO_NET_OR_BAD_PATH | Kein Netzwerkanbieter akzeptiert den angegebene Netzwerkpfad. |
| 1204 | 0x000004B4 | ERROR_BAD_PROVIDER | Der angegebene Netzwerk-Provider-Name ist ungültig. |
| 1205 | 0x000004B5 | ERROR_CANNOT_OPEN_PROFILE | Es kann nicht das Netzwerkverbindungsprofil geöffnet werden. |
| 1206 | 0x000004B6 | ERROR_BAD_PROFILE | Das Netzwerkverbindungsprofil ist beschädigt. |
| 1207 | 0x000004B7 | ERROR_NOT_CONTAINER | Eine der Nichtcontainer nicht aufgelistet werden. |
| 1208 | 0x000004B8 | ERROR_EXTENDED_ERROR | Ein erweiterter Fehler ist aufgetreten. |
| 1209 | 0x000004B9 | ERROR_INVALID_GROUPNAME | Das Format des angegebenen Gruppennamens ist ungültig. |
| 1210 | 0x000004BA | ERROR_INVALID_COMPUTERNAME | Das Format des angegebenen Computernamens ist ungültig. |
| 1211 | 0x000004BB | ERROR_INVALID_EVENTNAME | Das Format des angegebenen Ereignisnamens ist ungültig. |
| 1212 | 0x000004BC | ERROR_INVALID_DOMAINNAME | Das Format des angegebenen Domänennamens ist ungültig. |
| 1213 | 0x000004BD | ERROR_INVALID_SERVICENAME | Das Format des angegebenen Dienstnamens ist ungültig. |
| 1214 | 0x000004BE | ERROR_INVALID_NETNAME | Das Format des angegebenen Netzwerknamens ist ungültig. |
| 1215 | 0x000004BF | ERROR_INVALID_SHARENAME | Das Format des angegebenen Freigabenamens ist ungültig. |
| 1216 | 0x000004C0 | ERROR_INVALID_PASSWORDNAME | Das Format des angegebenen Kennworts ist ungültig. |
| 1217 | 0x000004C1 | ERROR_INVALID_MESSAGE_NAME | Das Format des angegebenen Meldung-Name ist ungültig. |
| 1218 | 0x000004C2 | ERROR_INVALID_MESSAGEDEST | Das Format des angegebenen Meldung Ziel ist ungültig. |
| 1219 | 0x000004C3 | ERROR_SESSION_CREDENTIAL_CONFLICT | Die Anmeldeinformationen angegeben Konflikt mit einem vorhandenen Satz von Anmeldeinformationen. |
| 1220 | 0x000004C4 | ERROR_REMOTE_SESSION_LIMIT_EXCEEDED | Es wurde versucht, eine Sitzung mit einem Netzwerkserver herzustellen, aber es gibt bereits zu viele Sitzungen mit diesem Server hergestellt. |
| 1221 | 0x000004C5 | ERROR_DUP_DOMAINNAME | Der Arbeitsgruppe oder Domäne-Name ist bereits von einem anderen Computer im Netzwerk verwendet. |
| 1222 | 0x000004C6 | ERROR_NO_NETWORK | Das Netzwerk ist nicht vorhanden oder nicht gestartet. |
| 1223 | 0x000004C7 | ERROR_CANCELLED | Der Vorgang wurde vom Benutzer abgebrochen. |
| 1224 | 0x000004C8 | ERROR_USER_MAPPED_FILE | Der angeforderte Vorgang kann nicht auf eine Datei mit einem Benutzer zugeordnet Abschnitt öffnen ausgeführt werden. |
| 1225 | 0x000004C9 | ERROR_CONNECTION_REFUSED | Das remote-System verweigert die Netzwerkverbindung. |
| 1226 | 0x000004CA | ERROR_GRACEFUL_DISCONNECT | Die Netzwerkverbindung wurde erfolgreich geschlossen. |
| 1227 | 0x000004CB | ERROR_ADDRESS_ALREADY_ASSOCIATED | Der Netzwerk-Transport-Endpunkt hat bereits eine Adresse zugeordnet. |
| 1228 | 0x000004CC | ERROR_ADDRESS_NOT_ASSOCIATED | Eine Adresse wurde noch nicht mit den Netzwerkendpunkt verbunden. |
| 1229 | 0x000004CD | ERROR_CONNECTION_INVALID | Ein Vorgang bezog sich auf eine nicht vorhandene Netzwerkverbindung. |
| 1230 | 0x000004CE | ERROR_CONNECTION_ACTIVE | Ein ungültiger Vorgang bezog sich auf eine aktive Netzwerkverbindung. |
| 1231 | 0x000004CF | ERROR_NETWORK_UNREACHABLE | Der Speicherort im Netzwerk nicht erreichbar. Informationen zur Fehlerbehebung in Netzwerken finden Sie in der Windows-Hilfe. |

| Fehler | | | Beschreibung |
|---------|-------------|-----------------------------------|---|
| Dezimal | Hexadezimal | Name | |
| 1232 | 0x000004D0 | ERROR_HOST_UNREACHABLE | Der Speicherort im Netzwerk nicht erreichbar. Informationen zur Fehlerbehebung in Netzwerken finden Sie in der Windows-Hilfe. |
| 1233 | 0x000004D1 | ERROR_PROTOCOL_UNREACHABLE | Der Speicherort im Netzwerk nicht erreichbar. Informationen zur Fehlerbehebung in Netzwerken finden Sie in der Windows-Hilfe. |
| 1234 | 0x000004D2 | ERROR_PORT_UNREACHABLE | Kein Dienst ist auf die Ziel-Netzwerk-Endpunkt auf dem entfernten System betrieben. |
| 1235 | 0x000004D3 | ERROR_REQUEST_ABORTED | Die Anforderung wurde abgebrochen. |
| 1236 | 0x000004D4 | ERROR_CONNECTION_ABORTED | Die Netzwerkverbindung wurde durch das lokale System abgebrochen. |
| 1237 | 0x000004D5 | ERROR_RETRY | Der Vorgang konnte nicht abgeschlossen werden. Eine Wiederholung sollte durchgeführt werden. |
| 1238 | 0x000004D6 | ERROR_CONNECTION_COUNT_LIMIT | Eine Verbindung zum Server konnte nicht erfolgen, da das Limit für die Anzahl der gleichzeitigen Verbindungen für dieses Konto erreicht wurde. |
| 1239 | 0x000004D7 | ERROR_LOGIN_TIME_RESTRICTION | Der Versuch, zu einer nicht autorisierten Tageszeit für dieses Konto einloggen. |
| 1240 | 0x000004D8 | ERROR_LOGIN_WKSTA_RESTRICTION | Das Konto ist nicht berechtigt, von dieser Station aus anmelden. |
| 1241 | 0x000004D9 | ERROR_INCORRECT_ADDRESS | Die Netzwerkadresse konnte für den angeforderten Vorgang nicht verwendet werden. |
| 1242 | 0x000004DA | ERROR_ALREADY_REGISTERED | Der Dienst ist bereits registriert. |
| 1243 | 0x000004DB | ERROR_SERVICE_NOT_FOUND | Der angegebene Dienst existiert nicht. |
| 1244 | 0x000004DC | ERROR_NOT_AUTHENTICATED | Der angeforderte Vorgang wurde nicht durchgeführt, weil der Benutzer nicht authentifiziert wurde. |
| 1245 | 0x000004DD | ERROR_NOT_LOGGED_ON | Der angeforderte Vorgang wurde nicht durchgeführt, weil der Benutzer nicht mit dem Netzwerk angemeldet hat. Der angegebene Dienst existiert nicht. |
| 1246 | 0x000004DE | ERROR_CONTINUE | Fahren Sie mit bei der Arbeit. |
| 1247 | 0x000004DF | ERROR_ALREADY_INITIALIZED | Es wurde versucht, einen Initialisierungsvorgang auszuführen, wenn die Initialisierung bereits abgeschlossen wurde. |
| 1248 | 0x000004E0 | ERROR_NO_MORE_DEVICES | Keine weiteren lokalen Geräte. |
| 1249 | 0x000004E1 | ERROR_NO_SUCH_SITE | Die angegebene Site ist nicht vorhanden. |
| 1250 | 0x000004E2 | ERROR_DOMAIN_CONTROLLER_EXISTS | Ein Domänencontroller mit dem angegebenen Namen ist bereits vorhanden. |
| 1251 | 0x000004E3 | ERROR_ONLY_IF_CONNECTED | Dieser Vorgang wird unterstützt, nur, wenn Sie mit dem Server verbunden sind. |
| 1252 | 0x000004E4 | ERROR_OVERRIDE_NOCHANGES | Die Gruppenrichtlinienumgebung sollte die Erweiterung aufrufen, auch wenn keine Änderungen vorliegen. |
| 1253 | 0x000004E5 | ERROR_BAD_USER_PROFILE | Der angegebene Benutzer muss kein gültiges Profil. |
| 1254 | 0x000004E6 | ERROR_NOT_SUPPORTED_ON_SBS | Dieser Vorgang wird auf einem Microsoft Small Business Server nicht unterstützt. |
| 1255 | 0x000004E7 | ERROR_SERVER_SHUTDOWN_IN_PROGRESS | Der Server-Computer wird heruntergefahren. |
| 1256 | 0x000004E8 | ERROR_HOST_DOWN | Das remote-System ist nicht verfügbar. Informationen zur Fehlerbehebung in Netzwerken finden Sie in der Windows-Hilfe. |
| 1257 | 0x000004E9 | ERROR_NON_ACCOUNT_SID | Der angegebene Sicherheitsbezeichner stammt nicht von einer Kontodomäne. |
| 1258 | 0x000004EA | ERROR_NON_DOMAIN_SID | Der angegebene Sicherheitsbezeichner verfügt über keine Domänenkomponente. |
| 1259 | 0x000004EB | ERROR_APPHELP_BLOCK | AppHelp Dialog abgebrochen, wodurch die Anwendung gestartet. |
| 1260 | 0x000004EC | ERROR_ACCESS_DISABLED_BY_POLICY | Zugriff auf die angeforderte Ressource wurde vom Systemadministrator deaktiviert. |
| 1261 | 0x000004ED | ERROR_REG_NAT_CONSUMPTION | Ein Programm-Versuch, verwenden Sie ein ungültiges registrieren Wert. Normalerweise verursacht durch einen nicht initialisierten Register. Dieser Fehler ist Itanium bestimmte. |
| 1262 | 0x000004EE | ERROR_CSCSHARE_OFFLINE | Die Freigabe ist derzeit offline oder existiert nicht. |

| Fehler | | | Beschreibung |
|---------|-------------|------------------------------|--|
| Dezimal | Hexadezimal | Name | |
| 1300 | 0x00000514 | ERROR_NOT_ALL_ASSIGNED | Nicht alle Privilegien verwiesen werden an den Aufrufer zugewiesen. |
| 1301 | 0x00000515 | ERROR_SOME_NOT_MAPPED | Einige Zuordnung von Kontennamen und Sicherheits-IDs nicht geschah. |
| 1302 | 0x00000516 | ERROR_NO_QUOTAS_FOR_ACCOUNT | Keine Kontingentgrenzen System sind für dieses Konto festgelegt worden. |
| 1303 | 0x00000517 | ERROR_LOCAL_USER_SESSION_KEY | Es ist kein Chiffrierschlüssel verfügbar. Ein bekannter Verschlüsselungsschlüssel wurde zurückgegeben. |
| 1304 | 0x00000518 | ERROR_NULL_LM_PASSWORD | Das Kennwort ist zu komplex, um in ein LAN Manager-Kennwort umgewandelt werden. Das zurückgegebene LAN Manager-Kennwort ist eine NULL-Zeichenfolge. |
| 1305 | 0x00000519 | ERROR_UNKNOWN_REVISION | Die Revisionsstufe ist nicht bekannt. |
| 1306 | 0x0000051A | ERROR_REVISION_MISMATCH | Gibt an, dass zwei Revisionsstufen nicht kompatibel sind. |
| 1307 | 0x0000051B | ERROR_INVALID_OWNER | Diese Sicherheits-ID kann nicht als Besitzer dieses Objekts zugewiesen werden. |
| 1308 | 0x0000051C | ERROR_INVALID_PRIMARY_GROUP | Diese Sicherheits-ID kann nicht als primäre Gruppe eines Objekts zugewiesen werden. |
| 1309 | 0x0000051D | ERROR_NO_IMPERSONATION_TOKEN | Eine hat versucht, ein Identitätstoken operieren von einem Thread, die einen Client derzeit keinen Identitätswechsel ist. |
| 1310 | 0x0000051E | ERROR_CANT_DISABLE_MANDATORY | Die Gruppe kann nicht deaktiviert werden. |
| 1311 | 0x0000051F | ERROR_NO_LOGON_SERVERS | Es sind momentan keine Anmeldeserver zur Verfügung, um die Anmeldeanforderung zu verarbeiten. |
| 1312 | 0x00000520 | ERROR_NO_SUCH_LOGON_SESSION | Eine angegebene Anmeldesitzung ist nicht vorhanden. Es kann bereits gekündigt haben. |
| 1313 | 0x00000521 | ERROR_NO_SUCH_PRIVILEGE | Angegebenen Privileg ist nicht vorhanden. |
| 1314 | 0x00000522 | ERROR_PRIVILEGE_NOT_HELD | Eine erforderliche Berechtigung ist nicht durch den Kunden statt. |
| 1315 | 0x00000523 | ERROR_INVALID_ACCOUNT_NAME | Der angegebene Name ist keine korrekt formatierte Kontonamen. |
| 1316 | 0x00000524 | ERROR_USER_EXISTS | Der angegebene Benutzer ist bereits vorhanden. |
| 1317 | 0x00000525 | ERROR_NO_SUCH_USER | Der angegebene Benutzer ist nicht vorhanden. |
| 1318 | 0x00000526 | ERROR_GROUP_EXISTS | Die angegebene Gruppe ist bereits vorhanden. |
| 1319 | 0x00000527 | ERROR_NO_SUCH_GROUP | Die angegebene Gruppe existiert nicht. |
| 1320 | 0x00000528 | ERROR_MEMBER_IN_GROUP | Entweder das angegebene Benutzerkonto ist bereits ein Mitglied der angegebenen Gruppe, oder die angegebene Gruppe kann nicht gelöscht werden, da sie ein Mitglied enthält. |
| 1321 | 0x00000529 | ERROR_MEMBER_NOT_IN_GROUP | Das angegebene Benutzerkonto ist kein Mitglied der angegebenen Gruppe Account. |
| 1322 | 0x0000052A | ERROR_LAST_ADMIN | Die letzte verbleibende Administratorkonto kann nicht deaktiviert oder gelöscht werden. |
| 1323 | 0x0000052B | ERROR_WRONG_PASSWORD | Nicht in der Lage, das Kennwort zu aktualisieren. Der Wert als das aktuelle Kennwort ist falsch. |
| 1324 | 0x0000052C | ERROR_ILL_FORMED_PASSWORD | Nicht in der Lage, das Kennwort zu aktualisieren. Für das neue Kennwort angegebene Wert enthält Werte, die in Kennwörtern nicht zulässig sind. |
| 1325 | 0x0000052D | ERROR_PASSWORD_RESTRICTION | Nicht in der Lage, das Kennwort zu aktualisieren. Der Wert für das neue Kennwort erfüllt nicht die Länge, der Komplexität oder der Geschichte Bedarf der Domäne. |
| 1326 | 0x0000052E | ERROR_LOGON_FAILURE | Anmeldung fehlgeschlagen: Unbekannter Benutzername oder falsches Kennwort. |
| 1327 | 0x0000052F | ERROR_ACCOUNT_RESTRICTION | Anmeldung fehlgeschlagen: Benutzerkontenbeschränkung. |
| 1328 | 0x00000530 | ERROR_INVALID_LOGON_HOURS | Anmeldung fehlgeschlagen: Konto-Anmeldung Zeit Einschränkung Verletzung. |
| 1329 | 0x00000531 | ERROR_INVALID_WORKSTATION | Anmeldung fehlgeschlagen: Benutzer nicht gestattet, an diesem Computer anmelden. |
| 1330 | 0x00000532 | ERROR_PASSWORD_EXPIRED | Anmeldung fehlgeschlagen: das angegebene Konto-Kennwort ist abgelaufen. |

| Fehler | | | Beschreibung |
|---------|-------------|--------------------------------|--|
| Dezimal | Hexadezimal | Name | |
| 1331 | 0x00000533 | ERROR_ACCOUNT_DISABLED | Anmeldung fehlgeschlagen: Konto zurzeit deaktiviert. |
| 1332 | 0x00000534 | ERROR_NONE_MAPPED | Zuordnungen von Kontennamen und Sicherheitskennungen wurden nicht durchgeführt. |
| 1333 | 0x00000535 | ERROR_TOO_MANY_LUIDS_REQUESTED | Zu viele lokale Benutzer-IDs (LUIDs) wurden gleichzeitig angefordert. |
| 1334 | 0x00000536 | ERROR_LUIDS_EXHAUSTED | Keine weiteren lokalen Benutzerkennungen (LUIDs) stehen zur Verfügung. |
| 1335 | 0x00000537 | ERROR_INVALID_SUB_AUTHORITY | Die Teilautoritätskomponente einer Sicherheits-ID ist ungültig für diese spezielle Verwendung. |
| 1336 | 0x00000538 | ERROR_INVALID_ACL | Die Struktur der Access Control List (ACL) ist ungültig. |
| 1337 | 0x00000539 | ERROR_INVALID_SID | Die Struktur der Sicherheits-ID ist ungültig. |
| 1338 | 0x0000053A | ERROR_INVALID_SECURITY_DESCR | Die Sicherheit Deskriptor Struktur ist ungültig. |
| 1340 | 0x0000053C | ERROR_BAD_INHERITANCE_ACL | Die vererbte Zugriffssteuerungsliste (ACL) oder Zugriffssteuerungseintrag (ACE) konnte nicht erstellt werden. |
| 1341 | 0x0000053D | ERROR_SERVER_DISABLED | Der Server ist derzeit deaktiviert. |
| 1342 | 0x0000053E | ERROR_SERVER_NOT_DISABLED | Der Server ist zurzeit aktiviert. |
| 1343 | 0x0000053F | ERROR_INVALID_ID_AUTHORITY | Der angegebene Wert wurde ein ungültiger Wert für Bezeichnerautorität. |
| 1344 | 0x00000540 | ERROR_ALLOTTED_SPACE_EXCEEDED | Kein Speicher mehr verfügbar ist für Sicherheits-Informationen-Updates verfügbar. |
| 1345 | 0x00000541 | ERROR_INVALID_GROUP_ATTRIBUTES | Die angegebenen Attribute sind ungültig oder nicht kompatibel mit den Attributen der Gruppe als Ganzes. |
| 1346 | 0x00000542 | ERROR_BAD_IMPERSONATION_LEVEL | Eine erforderliche Identitätswechselebene wurde nicht bereitgestellt oder die angegebene Identitätswechselebene ist ungültig. |
| 1347 | 0x00000543 | ERROR_CANT_OPEN_ANONYMOUS | Ein anonymer Sicherheitsebenentoken kann nicht geöffnet werden. |
| 1348 | 0x00000544 | ERROR_BAD_VALIDATION_CLASS | Die Validierung Informationsklasse angefordert war ungültig. |
| 1349 | 0x00000545 | ERROR_BAD_TOKEN_TYPE | Der Typ des Tokens ist für den versuchten Einsatz ungeeignet. |
| 1350 | 0x00000546 | ERROR_NO_SECURITY_ON_OBJECT | Nicht möglich eine Sicherheit für ein Objekt ohne zugeordnete Sicherheit aufweist. |
| 1351 | 0x00000547 | ERROR_CANT_ACCESS_DOMAIN_INFO | Konfigurationsinformationen konnte nicht vom Domänencontroller, gelesen werden da der Computer nicht verfügbar ist oder Zugriff wurde verweigert. |
| 1352 | 0x00000548 | ERROR_INVALID_SERVER_STATE | Die Sicherheitskontenverwaltung (SAM) oder lokale Sicherheits-Autorität (LSA) Server befand sich in den Zustand um die Sicherheit durchzuführen. |
| 1353 | 0x00000549 | ERROR_INVALID_DOMAIN_STATE | Die Domäne befand sich in einem Zustand, in den Sicherheitsvorgang auszuführen. |
| 1354 | 0x0000054A | ERROR_INVALID_DOMAIN_ROLE | Dieser Vorgang ist nur für den primären Domänencontroller der Domäne zulässig. |
| 1355 | 0x0000054B | ERROR_NO_SUCH_DOMAIN | Die angegebene Domäne ist nicht vorhanden oder konnte keine Verbindung hergestellt werden. |
| 1356 | 0x0000054C | ERROR_DOMAIN_EXISTS | Die angegebene Domäne ist bereits vorhanden. |
| 1357 | 0x0000054D | ERROR_DOMAIN_LIMIT_EXCEEDED | Wurde versucht, die Grenze für die Anzahl der Domänen je Server zu überschreiten. |
| 1358 | 0x0000054E | ERROR_INTERNAL_DB_CORRUPTION | Kann nicht die angeforderte Operation wegen einer katastrophalen Datenträgerfehler oder eine Beschädigung von Daten-Struktur auf der Festplatte abgeschlossen. |
| 1359 | 0x0000054F | ERROR_INTERNAL_ERROR | Ein interner Fehler aufgetreten. |
| 1360 | 0x00000550 | ERROR_GENERIC_NOT_MAPPED | Generische Zugriffarten wurden in einer Zugriffsmaske enthalten, die bereits nicht generischen Typen zugeordnet sein sollten. |
| 1361 | 0x00000551 | ERROR_BAD_DESCRIPTOR_FORMAT | Eine Sicherheitsbeschreibung ist nicht im richtigen Format (absolut oder selbstbezogen). |

| Fehler | | | Beschreibung |
|---------|-------------|------------------------------------|--|
| Dezimal | Hexadezimal | Name | |
| 1362 | 0x00000552 | ERROR_NOT_LOGON_PROCESS | Die angeforderte Aktion ist auf die Verwendung von Anmeldeprozessen nur beschränkt. Der aufrufende Prozess ist nicht als einen Anmeldevorgang registriert. |
| 1363 | 0x00000553 | ERROR_LOGON_SESSION_EXISTS | Eine neue Anmeldesitzung kann nicht mit der ID gestartet werden, die bereits verwendet wird. |
| 1364 | 0x00000554 | ERROR_NO_SUCH_PACKAGE | Ein angegebenen Authentifizierungspaket ist unbekannt. |
| 1365 | 0x00000555 | ERROR_BAD_LOGON_SESSION_STATE | Die Anmeldesitzung ist nicht in einem Staat, der die angeforderte Operation entspricht. |
| 1366 | 0x00000556 | ERROR_LOGON_SESSION_COLLISION | Die Anmeldesitzung, die, der ID bereits wird, verwendet. |
| 1367 | 0x00000557 | ERROR_INVALID_LOGON_TYPE | Eine Anmeldeanforderung enthielt einen Typwert ungültige Anmeldeversuche. |
| 1368 | 0x00000558 | ERROR_CANNOT_IMPERSONATE | Nicht mit mit einer named Pipe, bis Daten aus dem Rohr gelesen wurde. |
| 1369 | 0x00000559 | ERROR_RXACT_INVALID_STATE | Der Transaktionszustand einer Teilstruktur der Registrierung ist nicht kompatibel mit dem angeforderten Vorgang. |
| 1370 | 0x0000055A | ERROR_RXACT_COMMIT_FAILURE | Eine innere Sicherheit-Datenbankfehler ist aufgetreten. |
| 1371 | 0x0000055B | ERROR_SPECIAL_ACCOUNT | Diesen Vorgang für integrierte Konten nicht möglich. |
| 1372 | 0x0000055C | ERROR_SPECIAL_GROUP | Dieser Vorgang auf integrierte Gruppe kann nicht ausgeführt werden. |
| 1373 | 0x0000055D | ERROR_SPECIAL_USER | Dieser Vorgang auf diesen integrierten spezielle Benutzer kann nicht durchgeführt werden. |
| 1374 | 0x0000055E | ERROR_MEMBERS_PRIMARY_GROUP | Der Benutzer kann nicht aus einer Gruppe entfernt werden, da die Gruppe momentan die primäre Gruppe des Benutzers ist. |
| 1375 | 0x0000055F | ERROR_TOKEN_ALREADY_IN_USE | Das Token wird bereits als primäres Token verwendet. |
| 1376 | 0x00000560 | ERROR_NO_SUCH_ALIAS | Die angegebene lokale Gruppe ist nicht vorhanden. |
| 1377 | 0x00000561 | ERROR_MEMBER_NOT_IN_ALIAS | Der angegebene Kontenname ist kein Mitglied der lokalen Gruppe. |
| 1378 | 0x00000562 | ERROR_MEMBER_IN_ALIAS | Der angegebene Kontenname ist bereits Mitglied der lokalen Gruppe. |
| 1379 | 0x00000563 | ERROR_ALIAS_EXISTS | Die angegebene lokale Gruppe ist bereits vorhanden. |
| 1380 | 0x00000564 | ERROR_LOGON_NOT_GRANTED | Anmeldung fehlgeschlagen: der Benutzer wurde den angeforderte Anmeldetyp auf diesem Computer nicht erteilt. |
| 1381 | 0x00000565 | ERROR_TOO_MANY_SECRETS | Die maximale Anzahl der Geheimnisse, die in einem einzigen System gespeichert werden können, wurde überschritten. |
| 1382 | 0x00000566 | ERROR_SECRET_TOO_LONG | Die Länge eines Geheimnisses überschreitet die maximale zulässige Länge. |
| 1383 | 0x00000567 | ERROR_INTERNAL_DB_ERROR | Die lokale Sicherheits-Autorität-Datenbank enthält eine interne Inkonsistenz. |
| 1384 | 0x00000568 | ERROR_TOO_MANY_CONTEXT_IDS | Im Verlauf eines Anmeldeversuchs angesammelt Sicherheitskontext des Benutzers zu viele Sicherheits-IDs. |
| 1385 | 0x00000569 | ERROR_LOGON_TYPE_NOT_GRANTED | Anmeldung fehlgeschlagen: der Benutzer wurde den angeforderte Anmeldetyp auf diesem Computer nicht erteilt. |
| 1386 | 0x0000056A | ERROR_NT_CROSS_ENCRYPTION_REQUIRED | Ein querverschlüsseltes Kennwort ist erforderlich, ein Benutzerkennwort zu ändern. |
| 1387 | 0x0000056B | ERROR_NO_SUCH_MEMBER | Ein neues Mitglied konnte nicht hinzugefügt oder entfernt aus der lokalen Gruppe, da das Mitglied nicht vorhanden ist. |
| 1388 | 0x0000056C | ERROR_INVALID_MEMBER | Ein neues Mitglied konnte nicht zu einer lokalen Gruppe hinzugefügt werden, da das Mitglied den falschen Kontentyp aufweist. |
| 1389 | 0x0000056D | ERROR_TOO_MANY_SIDS | Zu viele Sicherheits-IDs es wurden angegeben. |

| Fehler | | | Beschreibung |
|---------|-------------|------------------------------------|---|
| Dezimal | Hexadezimal | Name | |
| 1390 | 0x0000056E | ERROR_LM_CROSS_ENCRYPTION_REQUIRED | Ein querverschlüsseltes Kennwort ist erforderlich, um dieses Benutzerkennwort zu ändern. |
| 1391 | 0x0000056F | ERROR_NO_INHERITANCE | Gibt an, dass eine Zugriffssteuerungsliste keine vererbaren Komponenten enthält. |
| 1392 | 0x00000570 | ERROR_FILE_CORRUPT | Die Datei oder das Verzeichnis ist beschädigt und nicht lesbar. |
| 1393 | 0x00000571 | ERROR_DISK_CORRUPT | Die Datenträgerstruktur ist beschädigt und nicht lesbar. |
| 1394 | 0x00000572 | ERROR_NO_USER_SESSION_KEY | Es ist kein Benutzersitzungsschlüssel für die angegebene Anmeldesitzung. |
| 1395 | 0x00000573 | ERROR_LICENSE_QUOTA_EXCEEDED | Der Dienst, auf die zugegriffen wird für eine bestimmte Anzahl von Verbindungen lizenziert. Keine weitere Verbindungen können zu dieser Zeit zum Dienst erfolgen, denn es gibt bereits so viele Verbindungen wie der Service annehmen kann. |
| 1396 | 0x00000574 | ERROR_WRONG_TARGET_NAME | Anmeldung fehlgeschlagen: Der Zielkontenname ist falsch. |
| 1397 | 0x00000575 | ERROR_MUTUAL_AUTH_FAILED | Gegenseitiger Authentifizierung ist fehlgeschlagen. Das Serverkennwort ist nicht mehr aktuell auf dem Domänencontroller. |
| 1398 | 0x00000576 | ERROR_TIME_SKEW | Gibt es einen Zeitunterschied zwischen Client und Server. |
| 1399 | 0x00000577 | ERROR_CURRENT_DOMAIN_NOT_ALLOWED | Dieser Vorgang kann nicht auf der aktuellen Domäne ausgeführt werden. |

| Fehler | | | Beschreibung |
|---------|-------------|--------------------------------------|---|
| Dezimal | Hexadezimal | Name | |
| 1400 | 0x00000578 | ERROR_INVALID_WINDOW_HANDLE | Ungültiges Fensterhandle. |
| 1401 | 0x00000579 | ERROR_INVALID_MENU_HANDLE | Ungültiges Menü-Handle. |
| 1402 | 0x0000057A | ERROR_INVALID_CURSOR_HANDLE | Ungültiger Cursor-Handle. |
| 1403 | 0x0000057B | ERROR_INVALID_ACCEL_HANDLE | Ungültige Beschleuniger Tabelle Handle. |
| 1404 | 0x0000057C | ERROR_INVALID_HOOK_HANDLE | Ungültiger Hook-Handle. |
| 1405 | 0x0000057D | ERROR_INVALID_DWP_HANDLE | Ungültiges Handle für eine Multiple-Fensterposition-Struktur. |
| 1406 | 0x0000057E | ERROR_TLW_WITH_WSCHILD | Auf der obersten Ebene untergeordnetes Fenster kann nicht erstellt werden. |
| 1407 | 0x0000057F | ERROR_CANNOT_FIND_WND_CLASS | Fensterklasse wurde nicht gefunden. |
| 1408 | 0x00000580 | ERROR_WINDOW_OF_OTHER_THREAD | Ungültiges Fenster; Es gehört zum anderen Thread. |
| 1409 | 0x00000581 | ERROR_HOTKEY_ALREADY_REGISTERED | Abkürzungstaste ist bereits registriert. |
| 1410 | 0x00000582 | ERROR_CLASS_ALREADY_EXISTS | Klasse ist bereits vorhanden. |
| 1411 | 0x00000583 | ERROR_CLASS_DOES_NOT_EXIST | Klasse ist nicht vorhanden. |
| 1412 | 0x00000584 | ERROR_CLASS_HAS_WINDOWS | Klasse hat noch geöffnete Fenster. |
| 1413 | 0x00000585 | ERROR_INVALID_INDEX | Ungültiger Index. |
| 1414 | 0x00000586 | ERROR_INVALID_ICON_HANDLE | Ungültiges Symbolhandle. |
| 1415 | 0x00000587 | ERROR_PRIVATE_DIALOG_INDEX | Private DIALOG-Fenster Wörter verwenden. |
| 1416 | 0x00000588 | ERROR_LISTBOX_ID_NOT_FOUND | Der Listenfeldbezeichner wurde nicht gefunden. |
| 1417 | 0x00000589 | ERROR_NO_WILDCARD_CHARACTERS | Es fanden sich keine Platzhalter. |
| 1418 | 0x0000058A | ERROR_CLIPBOARD_NOT_OPEN | Thread muss keine Zwischenablage öffnen. |
| 1419 | 0x0000058B | ERROR_HOTKEY_NOT_REGISTERED | Abkürzungstaste ist nicht registriert. |
| 1420 | 0x0000058C | ERROR_WINDOW_NOT_DIALOG | Das Fenster ist kein gültiges Dialogfenster. |
| 1421 | 0x0000058D | ERROR_CONTROL_ID_NOT_FOUND | Steuerelement-ID nicht gefunden. |
| 1422 | 0x0000058E | ERROR_INVALID_COMBOBOX_MESSAGE | Ungültige Nachricht für ein Kombinationsfeld, da es kein Edit-Steuerelement verfügt. |
| 1423 | 0x0000058F | ERROR_WINDOW_NOT_COMBOBOX | Das Fenster ist kein Kombinationsfeld. |
| 1424 | 0x00000590 | ERROR_INVALID_EDIT_HEIGHT | Höhe muss weniger als 256 sein. |
| 1425 | 0x00000591 | ERROR_DC_NOT_FOUND | Ungültiges Gerät Kontexthandle (DC). |
| 1426 | 0x00000592 | ERROR_INVALID_HOOK_FILTER | Ungültiger Hook-Prozedur-Typ. |
| 1427 | 0x00000593 | ERROR_INVALID_FILTER_PROC | Ungültiger Hook-Prozedur. |
| 1428 | 0x00000594 | ERROR_HOOK_NEEDS_HMOD | Kann nicht-lokaler Hook ohne ein Modulhandle gesetzt. |
| 1429 | 0x00000595 | ERROR_GLOBAL_ONLY_HOOK | Diese Hookprozedur kann nur global festgelegt werden. |
| 1430 | 0x00000596 | ERROR_JOURNAL_HOOK_SET | Die Journal-Hookprozedur ist bereits installiert. |
| 1431 | 0x00000597 | ERROR_HOOK_NOT_INSTALLED | Die Hookprozedur ist nicht installiert. |
| 1432 | 0x00000598 | ERROR_INVALID_LB_MESSAGE | Ungültige Nachricht für Listenfeld für Einfachauswahl. |
| 1433 | 0x00000599 | ERROR_SETCOUNT_ON_BAD_LB | LB_SETCOUNT an non-lazy-Listenfeld gesendet. |
| 1434 | 0x0000059A | ERROR_LB_WITHOUT_TABSTOPS | Dieses Listenfeld unterstützt keine Tabstopps. |
| 1435 | 0x0000059B | ERROR_DESTROY_OBJECT_OF_OTHER_THREAD | Von einem anderen Thread erstellte Objekt kann nicht zerstört werden. |
| 1436 | 0x0000059C | ERROR_CHILD_WINDOW_MENU | Untergeordnete Fenster können nicht Menüs haben. |
| 1437 | 0x0000059D | ERROR_NO_SYSTEM_MENU | Das Fenster muss kein Systemmenü. |
| 1438 | 0x0000059E | ERROR_INVALID_MSGBOX_STYLE | Ungültige Nachricht Knotenart. |
| 1439 | 0x0000059F | ERROR_INVALID_SPI_VALUE | Ungültige systemweite (SPI_*)-Parameter. |
| 1440 | 0x000005A0 | ERROR_SCREEN_ALREADY_LOCKED | Bildschirm bereits gesperrt. |
| 1441 | 0x000005A1 | ERROR_HWNDSDS_HAVE_DIFF_PARENT | Alle Handles für Windows in einer Multiple-Fenster-Position-Struktur müssen die gleichen Eltern. |
| 1442 | 0x000005A2 | ERROR_NOT_CHILD_WINDOW | Das Fenster ist kein untergeordnetes Fenster. |
| 1443 | 0x000005A3 | ERROR_INVALID_GW_COMMAND | Ungültiger GW_*-Befehl. |
| 1444 | 0x000005A4 | ERROR_INVALID_THREAD_ID | Ungültiger Threadbezeichner. |
| 1445 | 0x000005A5 | ERROR_NON_MDICHILD_WINDOW | Eine Nachricht aus einem Fenster, die kein mehrere Document Interface (MDI)-Fenster ist nicht verarbeitet werden. |
| 1446 | 0x000005A6 | ERROR_POPUP_ALREADY_ACTIVE | Popup-Menü bereits aktiv. |

| Fehler | | | Beschreibung |
|---------|-------------|--|---|
| Dezimal | Hexadezimal | Name | |
| 1447 | 0x000005A7 | ERROR_NO_SCROLLBARS | Das Fenster hat keine Bildlaufleisten. |
| 1448 | 0x000005A8 | ERROR_INVALID_SCROLLBAR_RANGE | Scroll Bar Bereich sein nicht größer als MAXLONG. |
| 1449 | 0x000005A9 | ERROR_INVALID_SHOWWIN_COMMAND | Nicht anzeigen oder entfernen das Fenster in der Art und Weise angegebene. |
| 1450 | 0x000005AA | ERROR_NO_SYSTEM_RESOURCES | Es gibt nicht genügend Systemressourcen um den angeforderten Dienst auszuführen. |
| 1451 | 0x000005AB | ERROR_NONPAGED_SYSTEM_RESOURCES | Es gibt nicht genügend Systemressourcen um den angeforderten Dienst auszuführen. |
| 1452 | 0x000005AC | ERROR_PAGED_SYSTEM_RESOURCES | Es gibt nicht genügend Systemressourcen um den angeforderten Dienst auszuführen. |
| 1453 | 0x000005AD | ERROR_WORKING_SET_QUOTA | Nicht genügend Quoten, um den angeforderten Dienst auszuführen. |
| 1454 | 0x000005AE | ERROR_PAGEFILE_QUOTA | Nicht genügend Quoten, um den angeforderten Dienst auszuführen. |
| 1455 | 0x000005AF | ERROR_COMMITMENT_LIMIT | Die Auslagerungsdatei ist zu klein für diesen Vorgang abgeschlossen. |
| 1456 | 0x000005B0 | ERROR_MENU_ITEM_NOT_FOUND | Ein Menüelement wurde nicht gefunden. |
| 1457 | 0x000005B1 | ERROR_INVALID_KEYBOARD_HANDLE | Ungültige Tastatur Layout Handle. |
| 1458 | 0x000005B2 | ERROR_HOOK_TYPE_NOT_ALLOWED | Hooktyp ist nicht zulässig. |
| 1459 | 0x000005B3 | ERROR_REQUIRES_INTERACTIVE_WINDOWSTATION | Dieser Vorgang erfordert eine interaktive Arbeitsstation. |
| 1460 | 0x000005B4 | ERROR_TIMEOUT | Dieser Vorgang zurückgegeben, da das Timeout ist abgelaufen. |
| 1461 | 0x000005B5 | ERROR_INVALID_MONITOR_HANDLE | Ungültige Monitor Handle. |
| 1500 | 0x000005DC | ERROR_EVENTLOG_FILE_CORRUPT | Die Ereignisprotokolldatei ist beschädigt. |
| 1501 | 0x000005DD | ERROR_EVENTLOG_CANT_START | Keine Ereignisprotokolldatei konnte geöffnet werden, so dass der Ereignis-Protokollierung-Dienst nicht gestartet werden konnte. |
| 1502 | 0x000005DE | ERROR_LOG_FILE_FULL | Die Ereignisprotokolldatei ist voll. |
| 1503 | 0x000005DF | ERROR_EVENTLOG_FILE_CHANGED | Die Ereignisprotokolldatei wurde zwischen den Lesevorgängen geändert. |

| Fehler | | | Beschreibung |
|---------|-------------|------------------------------------|--|
| Dezimal | Hexadezimal | Name | |
| 1601 | 0x00000641 | ERROR_INSTALL_SERVICE_FAILURE | Der Windows Installer-Dienst konnte nicht zugegriffen werden. Kontaktieren Sie den Support, um sicherzustellen, dass der Windows-Installationsdienst ordnungsgemäß registriert ist. |
| 1602 | 0x00000642 | ERROR_INSTALL_USEREXIT | Benutzer hat die Installation abgebrochen. |
| 1603 | 0x00000643 | ERROR_INSTALL_FAILURE | Schwerwiegender Fehler während der Installation. |
| 1604 | 0x00000644 | ERROR_INSTALL_SUSPEND | Installation angehalten, unvollständig. |
| 1605 | 0x00000645 | ERROR_UNKNOWN_PRODUCT | Diese Aktion gilt nur für Produkte, die derzeit installiert sind. |
| 1606 | 0x00000646 | ERROR_UNKNOWN_FEATURE | Funktionskennung ist nicht registriert. |
| 1607 | 0x00000647 | ERROR_UNKNOWN_COMPONENT | Komponentenkennung ist nicht registriert. |
| 1608 | 0x00000648 | ERROR_UNKNOWN_PROPERTY | Unbekannte Eigenschaft. |
| 1609 | 0x00000649 | ERROR_INVALID_HANDLE_STATE | Handle ist in einem ungültigen Zustand. |
| 1610 | 0x0000064A | ERROR_BAD_CONFIGURATION | Die Konfigurationsdaten für dieses Produkt ist beschädigt. Kontaktieren Sie Ihren Support. |
| 1611 | 0x0000064B | ERROR_INDEX_ABSENT | Komponente-Qualifizierer ist nicht vorhanden. |
| 1612 | 0x0000064C | ERROR_INSTALL_SOURCE_ABSENT | Die Installationsquelle für dieses Produkt ist nicht verfügbar. Stellen Sie sicher, dass die Quelle vorhanden ist und dass Sie darauf zugreifen können. |
| 1613 | 0x0000064D | ERROR_INSTALL_PACKAGE_VERSION | Dieses Installationspaket kann nicht vom Windows-Installationsdienst installiert werden. Sie müssen ein Windows Servicepack installieren, das eine neuere Version des Windows Installer-Dienstes enthält. |
| 1614 | 0x0000064E | ERROR_PRODUCT_UNINSTALLED | Produkt deinstalliert wird. |
| 1615 | 0x0000064F | ERROR_BAD_QUERY_SYNTAX | SQL-Abfragesyntax ungültig oder wird nicht unterstützt. |
| 1616 | 0x00000650 | ERROR_INVALID_FIELD | Eintragsfeld ist nicht vorhanden. |
| 1617 | 0x00000651 | ERROR_DEVICE_REMOVED | Das Gerät wurde entfernt. |
| 1618 | 0x00000652 | ERROR_INSTALL_ALREADY_RUNNING | Eine andere Installation wird bereits ausgeführt. Schließen Sie die Installation bevor Sie mit der Installation fortfahren. |
| 1619 | 0x00000653 | ERROR_INSTALL_PACKAGE_OPEN_FAILED | Dieses Installationspaket konnte nicht geöffnet werden. Stellen Sie sicher, dass das Paket vorhanden ist und dass Sie darauf zugreifen können, oder wenden Sie sich an den Anwendungshersteller, um sicherzustellen, dass dies ein gültiges Windows Installer-Paket ist. |
| 1620 | 0x00000654 | ERROR_INSTALL_PACKAGE_INVALID | Dieses Installationspaket konnte nicht geöffnet werden. Wenden Sie sich an den Anwendungshersteller, um sicherzustellen, dass dies ein gültiges Windows Installer-Paket ist. |
| 1621 | 0x00000655 | ERROR_INSTALL_UI_FAILURE | Es gab einen Fehler beim Starten der Benutzeroberfläche des Windows Installer-Dienst. Kontaktieren Sie Ihren Support. |
| 1622 | 0x00000656 | ERROR_INSTALL_LOG_FAILURE | Fehler beim Öffnen der Installationsprotokolldatei. Stellen Sie sicher, dass der Speicherort der angegebenen Protokolldatei vorhanden ist und dass Sie darauf schreiben können. |
| 1623 | 0x00000657 | ERROR_INSTALL_LANGUAGE_UNSUPPORTED | Die Sprache dieses Installationspakets wird von Ihrem System nicht unterstützt. |
| 1624 | 0x00000658 | ERROR_INSTALL_TRANSFORM_FAILURE | Fehler beim Anwenden der Transformationen. Stellen Sie sicher, dass die angegebenen Transformationspfade gültig sind. |
| 1625 | 0x00000659 | ERROR_INSTALL_PACKAGE_REJECTED | Diese Installation ist von Systemrichtlinien verboten. Kontaktieren Sie Ihren Systemadministrator. |
| 1626 | 0x0000065A | ERROR_FUNCTION_NOT_CALLED | Funktion konnte nicht ausgeführt werden. |
| 1627 | 0x0000065B | ERROR_FUNCTION_FAILED | Funktion ist während der Ausführung fehlgeschlagen. |
| 1628 | 0x0000065C | ERROR_INVALID_TABLE | Ungültige oder unbekannte Tabelle angegeben. |
| 1629 | 0x0000065D | ERROR_DATATYPE_MISMATCH | Bereitgestellten Daten sind vom falschen Typ. |
| 1630 | 0x0000065E | ERROR_UNSUPPORTED_TYPE | Daten dieser Art werden nicht unterstützt. |

| Fehler | | | Beschreibung |
|---------|-------------|------------------------------------|---|
| Dezimal | Hexadezimal | Name | |
| 1631 | 0x0000065F | ERROR_CREATE_FAILED | Der Windows Installer-Dienst konnte nicht gestartet werden. Kontaktieren Sie Ihren Support. |
| 1632 | 0x00000660 | ERROR_INSTALL_TEMP_UNWRITABLE | Der temp-Ordner ist entweder voll oder unzugänglich. Überprüfen Sie, ob der temp-Ordner vorhanden ist und ob Sie darauf schreiben können. |
| 1633 | 0x00000661 | ERROR_INSTALL_PLATFORM_UNSUPPORTED | Dieses Installationspaket wird von diesem Prozessortyp nicht unterstützt. Wenden Sie sich an Ihren Hersteller. |
| 1634 | 0x00000662 | ERROR_INSTALL_NOTUSED | Komponente auf diesem Computer nicht verwendet. |
| 1635 | 0x00000663 | ERROR_PATCH_PACKAGE_OPEN_FAILED | Das Patchpaket konnte nicht geöffnet werden. Stellen Sie sicher, dass das Patchpaket vorhanden ist und dass Sie darauf zugreifen können, oder wenden Sie sich an den Anwendungshersteller, um sicherzustellen, dass dies ein gültiges Windows Installer-Patchpaket ist. |
| 1636 | 0x00000664 | ERROR_PATCH_PACKAGE_INVALID | Das Patchpaket konnte nicht geöffnet werden. Wenden Sie sich an den Anwendungshersteller, um sicherzustellen, dass dies ein gültiges Windows Installer-Patchpaket ist. |
| 1637 | 0x00000665 | ERROR_PATCH_PACKAGE_UNSUPPORTED. | Dieses Patchpaket kann vom Windows-Installationsdienst nicht verarbeitet werden. Sie müssen ein Windows Servicepack installieren, das eine neuere Version des Windows Installer-Dienstes enthält. |
| 1638 | 0x00000666 | ERROR_PRODUCT_VERSION | Eine andere Version dieses Produkts ist bereits installiert. Installation dieser Version kann nicht fortgesetzt werden. Zum Konfigurieren oder die vorhandene Version dieses Produkts entfernen, verwenden Sie Programme hinzufügen/entfernen in der Systemsteuerung. |
| 1639 | 0x00000667 | ERROR_INVALID_COMMAND_LINE | Ungültiges Befehlszeilenargument. Ausführliche Befehlszeilenhilfe finden Sie im Windows Installer SDK. |
| 1640 | 0x00000668 | ERROR_INSTALL_REMOTE_DISALLOWED | Nur Administratoren haben die Berechtigung zum Hinzufügen, Entfernen oder während einer Terminaldienste-Remotesitzung Serversoftware konfigurieren. Wenn Sie installieren oder Software auf dem Server konfigurieren möchten, wenden Sie sich an den Netzwerkadministrator. |
| 1641 | 0x00000669 | ERROR_SUCCESS_REBOOT_INITIATED | Der angeforderte Vorgang wurde erfolgreich abgeschlossen. Das System wird neu gestartet werden, damit die Änderungen wirksam werden. |
| 1642 | 0x0000066A | ERROR_PATCH_TARGET_NOT_FOUND | Das Updatepatch kann nicht vom Windows-Installationsdienst installiert werden, da das Programm aktualisiert werden, fehlen möglicherweise, oder das Updatepatch kann eine andere Version des Programms aktualisieren. Stellen Sie sicher, dass das Programm aktualisiert werden auf Ihrem Computer vorhanden ist und dass Sie das richtige Updatepatch. |
| 1643 | 0x0000066B | ERROR_PATCH_PACKAGE_REJECTED | Das Patchpaket ist von der Systemrichtlinie nicht gestattet. Es ist nicht mit einem entsprechenden Zertifikat signiert. |
| 1644 | 0x0000066C | ERROR_INSTALL_TRANSFORM_REJECTED | Eine oder mehrere Anpassungen sind von der Systemrichtlinie nicht zulässig. Sie werden nicht mit einem entsprechenden Zertifikat signiert. |
| 1700 | 0x000006A4 | RPC_S_INVALID_STRING_BINDING | Die String-Bindung ist ungültig. |
| 1701 | 0x000006A5 | RPC_S_WRONG_KIND_OF_BINDING | Die Bindungs-Zugriffsnummer hat den falschen Typ. |
| 1702 | 0x000006A6 | RPC_S_INVALID_BINDING | Die Bindungs-Zugriffsnummer ist ungültig. |
| 1703 | 0x000006A7 | RPC_S_PROTSEQ_NOT_SUPPORTED | Die RPC-Protokollsequenz wird nicht unterstützt. |
| 1704 | 0x000006A8 | RPC_S_INVALID_RPC_PROTSEQ | Die RPC-Protokollsequenz ist ungültig. |
| 1705 | 0x000006A9 | RPC_S_INVALID_STRING_UUID | Der Zeichenfolge universelle eindeutige Bezeichner (UUID) ist ungültig. |
| 1706 | 0x000006AA | RPC_S_INVALID_ENDPOINT_FORMAT | Das Endpunktformat ist ungültig. |
| 1707 | 0x000006AB | RPC_S_INVALID_NET_ADDR | Die Netzwerkadresse ist ungültig. |

| Fehler | | | Beschreibung |
|---------|-------------|-------------------------------|---|
| Dezimal | Hexadezimal | Name | |
| 1708 | 0x000006AC | RPC_S_NO_ENDPOINT_FOUND | Kein Endpunkt es wurde gefunden. |
| 1709 | 0x000006AD | RPC_S_INVALID_TIMEOUT | Der Timeoutwert ist ungültig. |
| 1710 | 0x000006AE | RPC_S_OBJECT_NOT_FOUND | Der Objekt universelle eindeutige Bezeichner (UUID) wurde nicht gefunden. |
| 1711 | 0x000006AF | RPC_S_ALREADY_REGISTERED | Der Objekt universelle eindeutige Bezeichner (UUID) wurde bereits registriert. |
| 1712 | 0x000006B0 | RPC_S_TYPE_ALREADY_REGISTERED | Der Typ universelle eindeutige Bezeichner (UUID) wurde bereits registriert. |
| 1713 | 0x000006B1 | RPC_S_ALREADY_LISTENING | Der RPC-Server ist bereits überwacht. |
| 1714 | 0x000006B2 | RPC_S_NO_PROTSEQS_REGISTERED | Keine Protokollsequenzen wurden registriert. |
| 1715 | 0x000006B3 | RPC_S_NOT_LISTENING | Der RPC-Server hört nicht. |
| 1716 | 0x000006B4 | RPC_S_UNKNOWN_MGR_TYPE | Der Managertyp ist unbekannt. |
| 1717 | 0x000006B5 | RPC_S_UNKNOWN_IF | Die Schnittstelle ist unbekannt. |
| 1718 | 0x000006B6 | RPC_S_NO_BINDINGS | Es gibt keine Bindungen. |
| 1719 | 0x000006B7 | RPC_S_NO_PROTSEQS | Es gibt keine Protokollsequenzen. |
| 1720 | 0x000006B8 | RPC_S_CANT_CREATE_ENDPOINT | Der Endpunkt kann nicht erstellt werden. |
| 1721 | 0x000006B9 | RPC_S_OUT_OF_RESOURCES | Sind nicht genügend Ressourcen zum Abschließen dieses Vorgangs verfügbar. |
| 1722 | 0x000006BA | RPC_S_SERVER_UNAVAILABLE | Der RPC-Server ist nicht verfügbar. |
| 1723 | 0x000006BB | RPC_S_SERVER_TOO_BUSY | Der RPC-Server ist ausgelastet, um diesen Vorgang abzuschließen. |
| 1724 | 0x000006BC | RPC_S_INVALID_NETWORK_OPTIONS | Die Netzwerkoptionen sind ungültig. |
| 1725 | 0x000006BD | RPC_S_NO_CALL_ACTIVE | Es sind keine Remoteprozeduraufrufe in diesem Thread aktiv. |
| 1726 | 0x000006BE | RPC_S_CALL_FAILED | Der Remoteprozeduraufruf ist fehlgeschlagen. |
| 1727 | 0x000006BF | RPC_S_CALL_FAILED_DNE | Der Remoteprozeduraufruf ist fehlgeschlagen und wurde nicht ausgeführt. |
| 1728 | 0x000006C0 | RPC_S_PROTOCOL_ERROR | Ein remote Prozedur Call (RPC)-Protokollfehler ist aufgetreten. |
| 1730 | 0x000006C2 | RPC_S_UNSUPPORTED_TRANS_SYN | Die Transfer-Syntax wird von der RPC-Server nicht unterstützt. |
| 1732 | 0x000006C4 | RPC_S_UNSUPPORTED_TYPE | Der universelle eindeutige Bezeichner (UUID)-Typ wird nicht unterstützt. |
| 1733 | 0x000006C5 | RPC_S_INVALID_TAG | Das Tag ist ungültig. |
| 1734 | 0x000006C6 | RPC_S_INVALID_BOUND | Die Datenfeldgrenzen sind ungültig. |
| 1735 | 0x000006C7 | RPC_S_NO_ENTRY_NAME | Die Bindung enthält keinem Eintragsnamen. |
| 1736 | 0x000006C8 | RPC_S_INVALID_NAME_SYNTAX | Die Namenssyntax ist ungültig. |
| 1737 | 0x000006C9 | RPC_S_UNSUPPORTED_NAME_SYNTAX | Die Namenssyntax wird nicht unterstützt. |
| 1739 | 0x000006CB | RPC_S_UUID_NO_ADDRESS | Keine Netzwerkadresse steht verwenden, um eine universelle eindeutige Bezeichner (UUID) zu erstellen. |
| 1740 | 0x000006CC | RPC_S_DUPLICATE_ENDPOINT | Der Endpunkt ist ein Duplikat. |
| 1741 | 0x000006CD | RPC_S_UNKNOWN_AUTHN_TYPE | Der Authentifizierungstyp ist unbekannt. |
| 1742 | 0x000006CE | RPC_S_MAX_CALLS_TOO_SMALL | Die maximale Anzahl der Aufrufe ist zu klein. |
| 1743 | 0x000006CF | RPC_S_STRING_TOO_LONG | Die Zeichenfolge ist zu lang. |
| 1744 | 0x000006D0 | RPC_S_PROTSEQ_NOT_FOUND | Die RPC-Protokollsequenz wurde nicht gefunden. |
| 1745 | 0x000006D1 | RPC_S_PROCNUM_OUT_OF_RANGE | Die Prozedur ist außerhalb des gültigen Bereichs. |
| 1746 | 0x000006D2 | RPC_S_BINDING_HAS_NO_AUTH | Die Bindung enthält keine Authentifizierungsinformationen nicht. |
| 1747 | 0x000006D3 | RPC_S_UNKNOWN_AUTHN_SERVICE | Der Echtheitsbestätigungsdienst ist unbekannt. |
| 1748 | 0x000006D4 | RPC_S_UNKNOWN_AUTHN_LEVEL | Die Authentifizierungsebene ist unbekannt. |
| 1749 | 0x000006D5 | RPC_S_INVALID_AUTH_IDENTITY | Der Sicherheitskontext ist ungültig. |
| 1750 | 0x000006D6 | RPC_S_UNKNOWN_AUTHZ_SERVICE | Der Autorisierungsdienst ist unbekannt. |
| 1751 | 0x000006D7 | EPT_S_INVALID_ENTRY | Der Eintrag ist ungültig. |
| 1752 | 0x000006D8 | EPT_S_CANT_PERFORM_OP | Der Serverendpunkt kann den Vorgang nicht ausführen. |
| 1753 | 0x000006D9 | EPT_S_NOT_REGISTERED | Es sind keine weiteren Endpunkte verfügbar in der Endpunktzusammenfassung. |
| 1754 | 0x000006DA | RPC_S_NOTHING_TO_EXPORT | Keine Schnittstellen wurden exportiert. |
| 1755 | 0x000006DB | RPC_S_INCOMPLETE_NAME | Der Eintragsname ist unvollständig. |
| 1756 | 0x000006DC | RPC_S_INVALID_VERS_OPTION | Die Versionsoption ist ungültig. |

| Fehler | | | Beschreibung |
|---------|-------------|------------------------------------|--|
| Dezimal | Hexadezimal | Name | |
| 1757 | 0x000006DD | RPC_S_NO_MORE_MEMBERS | Es gibt keine weitere Mitglieder. |
| 1758 | 0x000006DE | RPC_S_NOT_ALL_OBJS_UNEXPORTED | Es gibt nichts zu unexport. |
| 1759 | 0x000006DF | RPC_S_INTERFACE_NOT_FOUND | Die Schnittstelle wurde nicht gefunden. |
| 1760 | 0x000006E0 | RPC_S_ENTRY_ALREADY_EXISTS | Der Eintrag ist bereits vorhanden. |
| 1761 | 0x000006E1 | RPC_S_ENTRY_NOT_FOUND | Der Eintrag wurde nicht gefunden. |
| 1762 | 0x000006E2 | RPC_S_NAME_SERVICE_UNAVAILABLE | Der Namensdienst ist nicht verfügbar. |
| 1763 | 0x000006E3 | RPC_S_INVALID_NAF_ID | Die Familie der Netzwerk-Adresse ist ungültig. |
| 1764 | 0x000006E4 | RPC_S_CANNOT_SUPPORT | Der angeforderte Vorgang wird nicht unterstützt. |
| 1765 | 0x000006E5 | RPC_S_NO_CONTEXT_AVAILABLE | Kein Sicherheitskontext steht Identitätswechsel zu erlauben. |
| 1766 | 0x000006E6 | RPC_S_INTERNAL_ERROR | Ein Remoteprozeduraufruf (RPC) ist ein interner Fehler aufgetreten. |
| 1767 | 0x000006E7 | RPC_S_ZERO_DIVIDE | Der RPC-Server versuchte eine Ganzzahldivision durch Null. |
| 1768 | 0x000006E8 | RPC_S_ADDRESS_ERROR | Adressierung Fehler im RPC-Server. |
| 1769 | 0x000006E9 | RPC_S_FP_DIV_ZERO | Eine Gleitkommaoperation beim RPC-Server verursachte eine Division durch Null. |
| 1770 | 0x000006EA | RPC_S_FP_UNDERFLOW | Ein Gleitkommaunterlauf trat beim RPC-Server. |
| 1771 | 0x000006EB | RPC_S_FP_OVERFLOW | Ein Gleitkommaüberlauf trat beim RPC-Server. |
| 1772 | 0x000006EC | RPC_X_NO_MORE_ENTRIES | Die Liste der RPC-Server für die Bindung von Auto-Handles ist aufgebraucht. |
| 1773 | 0x000006ED | RPC_X_SS_CHAR_TRANS_OPEN_FAIL | Kann nicht die Tabelle Zeichenübersetzungsdatei geöffnet. |
| 1774 | 0x000006EE | RPC_X_SS_CHAR_TRANS_SHORT_FILE | Die Datei mit der Zeichenübersetzungstabelle hat weniger als 512 Bytes. |
| 1775 | 0x000006EF | RPC_X_SS_IN_NULL_CONTEXT | Ein null-Kontexthandle wurde während eines Remoteprozeduraufrufs vom Client an den Host übergeben. |
| 1777 | 0x000006F1 | RPC_X_SS_CONTEXT_DAMAGED | Das Kontexthandle während eines Remoteprozeduraufrufs geändert. |
| 1778 | 0x000006F2 | RPC_X_SS_HANDLES_MISMATCH | Die Bindungszugriffsnummern, die an einen entfernten Prozeduraufruf übergeben wurden, stimmen nicht überein. |
| 1779 | 0x000006F3 | RPC_X_SS_CANNOT_GET_CALL_HANDLE | Die Stub kann das remote Prozedur Call-Handle zu erhalten. |
| 1780 | 0x000006F4 | RPC_X_NULL_REF_POINTER | An den Stub wurde ein null-Verweis-Zeiger übergeben. |
| 1781 | 0x000006F5 | RPC_X_DEUM_VALUE_OUT_OF_RANGE | Der Enumerationswert ist außerhalb des gültigen Bereichs. |
| 1782 | 0x000006F6 | RPC_X_BYTE_COUNT_TOO_SMALL | Die Byteanzahl ist zu klein. |
| 1783 | 0x000006F7 | RPC_X_BAD_STUB_DATA | Die Stub erhielt fehlerhafte Daten. |
| 1784 | 0x000006F8 | ERROR_INVALID_USER_BUFFER | Der angegebene Benutzerpuffer ist für den angeforderten Vorgang ungültig. |
| 1785 | 0x000006F9 | ERROR_UNRECOGNIZED_MEDIA | Dem Speichermedium wird nicht erkannt. Es kann nicht formatiert werden. |
| 1786 | 0x000006FA | ERROR_NO_TRUST_LSA_SECRET | Die Arbeitsstation muss kein Geheimnis Vertrauen. |
| 1787 | 0x000006FB | ERROR_NO_TRUST_SAM_ACCOUNT | Die Sicherheitsdatenbank auf dem Server muss kein Computerkonto für diese Arbeitsstationsvertrauensstellung. |
| 1788 | 0x000006FC | ERROR_TRUSTED_DOMAIN_FAILURE | Die Vertrauensstellung zwischen der primären Domäne und der vertrauenswürdigen Domäne scheiterte. |
| 1789 | 0x000006FD | ERROR_TRUSTED_RELATIONSHIP_FAILURE | Die Vertrauensstellung zwischen dieser Arbeitsstation und der primären Domäne scheiterte. |
| 1790 | 0x000006FE | ERROR_TRUST_FAILURE | Die Netzwerkanmeldung ist fehlgeschlagen. |
| 1791 | 0x000006FF | RPC_S_CALL_IN_PROGRESS | Fortschritt für diesen Thread wird bereits ein remote Procedure Call. |
| 1792 | 0x00000700 | ERROR_NETLOGON_NOT_STARTED | Es wurde versucht, sich anzumelden, aber der Netzwerk Logon-Dienst wurde nicht gestartet. |
| 1793 | 0x00000701 | ERROR_ACCOUNT_EXPIRED | Das Benutzerkonto ist abgelaufen. |
| 1794 | 0x00000702 | ERROR_REDIRECTOR_HAS_OPEN_HANDLES | Der Redirector wird verwendet und kann nicht entladen werden. |

| Fehler | | | Beschreibung |
|---------|-------------|--|--|
| Dezimal | Hexadezimal | Name | |
| 1795 | 0x00000703 | ERROR_PRINTER_DRIVER_ALREADY_INSTALLED | Der angegebene Druckertreiber ist bereits installiert. |
| 1796 | 0x00000704 | ERROR_UNKNOWN_PORT | Der angegebene Anschluss ist unbekannt. |
| 1797 | 0x00000705 | ERROR_UNKNOWN_PRINTER_DRIVER | Der Druckertreiber ist unbekannt. |
| 1798 | 0x00000706 | ERROR_UNKNOWN_PRINTPROCESSOR | Der Druckprozessor ist unbekannt. |
| 1799 | 0x00000707 | ERROR_INVALID_SEPARATOR_FILE | Die angegebene Trenndatei ist ungültig. |

| Fehler | | | Beschreibung |
|---------|-------------|---|--|
| Dezimal | Hexadezimal | Name | |
| 1800 | 0x00000708 | ERROR_INVALID_PRIORITY | Die angegebene Priorität ist ungültig. |
| 1801 | 0x00000709 | ERROR_INVALID_PRINTER_NAME | Der Druckername ist ungültig. |
| 1802 | 0x0000070A | ERROR_PRINTER_ALREADY_EXISTS | Der Drucker ist bereits vorhanden. |
| 1803 | 0x0000070B | ERROR_INVALID_PRINTER_COMMAND | Der Druckerbefehl ist ungültig. |
| 1804 | 0x0000070C | ERROR_INVALID_DATATYPE | Der angegebene Datentyp ist ungültig. |
| 1805 | 0x0000070D | ERROR_INVALID_ENVIRONMENT | Die angegebene Umgebung ist ungültig. |
| 1806 | 0x0000070E | RPC_S_NO_MORE_BINDINGS | Es gibt keine weitere Bindungen. |
| 1807 | 0x0000070F | ERROR_NOLOGON_INTERDOMAIN_TRUST_ACCOUNT | Das verwendete Konto ist ein domänenübergreifendes Vertrauenskonto. Verwenden Sie Ihr globales Benutzerkonto oder lokales Benutzerkonto auf diesen Server zugreifen. |
| 1808 | 0x00000710 | ERROR_NOLOGON_WORKSTATION_TRUST_ACCOUNT | Das verwendete Konto ist ein Arbeitsstationsvertrauenskonto. Verwenden Sie Ihr globales Benutzerkonto oder lokales Benutzerkonto auf diesen Server zugreifen. |
| 1809 | 0x00000711 | ERROR_NOLOGON_SERVER_TRUST_ACCOUNT | Das verwendete Konto ist ein Serververtrauenskonto. Verwenden Sie Ihr globales Benutzerkonto oder lokales Benutzerkonto auf diesen Server zugreifen. |
| 1810 | 0x00000712 | ERROR_DOMAIN_TRUST_INCONSISTENT | Der Name oder die Sicherheitskennung (SID) der angegebenen Domäne ist unvereinbar mit den Vertrauensinformationen dieser Domäne. |
| 1811 | 0x00000713 | ERROR_SERVER_HAS_OPEN_HANDLES | Der Server wird verwendet und kann nicht entladen werden. |
| 1812 | 0x00000714 | ERROR_RESOURCE_DATA_NOT_FOUND | Die angegebene Bilddatei enthält keinen Ressourcenabschnitt. |
| 1813 | 0x00000715 | ERROR_RESOURCE_TYPE_NOT_FOUND | Der angegebene Ressourcentyp kann nicht in der Bilddatei gefunden werden. |
| 1814 | 0x00000716 | ERROR_RESOURCE_NAME_NOT_FOUND | Der angegebene Ressourcenname kann nicht in der Bilddatei gefunden werden. |
| 1815 | 0x00000717 | ERROR_RESOURCE_LANG_NOT_FOUND | Die angegebene Ressource-Sprach-ID kann nicht in der Bilddatei gefunden werden. |
| 1816 | 0x00000718 | ERROR_NOT_ENOUGH_QUOTA | Nicht genug Quote ist verfügbar, um diesen Befehl zu verarbeiten. |
| 1817 | 0x00000719 | RPC_S_NO_INTERFACES | Keine Schnittstellen wurden registriert. |
| 1818 | 0x0000071A | RPC_S_CALL_CANCELLED | Der Remoteprozeduraufruf wurde abgesagt. |
| 1819 | 0x0000071B | RPC_S_BINDING_INCOMPLETE | Die Bindungsnummer enthält nicht alle benötigten Informationen. |
| 1820 | 0x0000071C | RPC_S_COMM_FAILURE | Während ein remote Procedure Call ist ein Kommunikationsfehler aufgetreten. |
| 1821 | 0x0000071D | RPC_S_UNSUPPORTED_AUTHN_LEVEL | Die angeforderte Echtheitsbestätigungsebene wird nicht unterstützt. |
| 1822 | 0x0000071E | RPC_S_NO_PRINC_NAME | Kein Prinzipalname registriert. |
| 1823 | 0x0000071F | RPC_S_NOT_RPC_ERROR | Der angegebene Fehler ist kein gültiger Windows-RPC Fehler-Code. |
| 1824 | 0x00000720 | RPC_S_UUID_LOCAL_ONLY | Eine UUID, die nur für diesen Computer gültig ist wurde zugewiesen. |
| 1825 | 0x00000721 | RPC_S_SEC_PKG_ERROR | Ein Paket spezifischen Sicherheitsfehler. |
| 1826 | 0x00000722 | RPC_S_NOT_CANCELLED | Thread wird nicht abgebrochen. |
| 1827 | 0x00000723 | RPC_X_INVALID_ES_ACTION | Ungültiger Vorgang auf dem Handle Kodierung/Dekodierung. |
| 1828 | 0x00000724 | RPC_X_WRONG_ES_VERSION | Inkompatible Version des Pakets serialisieren. |
| 1829 | 0x00000725 | RPC_X_WRONG_STUB_VERSION | Inkompatible Version des RPC-Stub. |
| 1830 | 0x00000726 | RPC_X_INVALID_PIPE_OBJECT | Das RPC-Pipe-Objekt ist ungültig oder beschädigt. |
| 1831 | 0x00000727 | RPC_X_WRONG_PIPE_ORDER | Ein ungültiger Vorgang bezog sich auf eine RPC-Pipe-Objekt. |
| 1832 | 0x00000728 | RPC_X_WRONG_PIPE_VERSION | Nicht unterstützte RPC-Pipe-Version. |
| 1898 | 0x0000076A | RPC_S_GROUP_MEMBER_NOT_FOUND | Mitglied der Gruppe wurde nicht gefunden. |
| 1899 | 0x0000076B | EPT_S_CANT_CREATE | Der Endpunkt-Mapper-Datenbank-Eintrag konnte nicht erstellt werden. |
| 1900 | 0x0000076C | RPC_S_INVALID_OBJECT | Der Objekt universelle eindeutige Bezeichner (UUID) ist die nil-UUID. |

| Fehler | | | Beschreibung |
|---------|-------------|-----------------------------------|---|
| Dezimal | Hexadezimal | Name | |
| 1901 | 0x0000076D | ERROR_INVALID_TIME | Die angegebene Zeit ist ungültig. |
| 1902 | 0x0000076E | ERROR_INVALID_FORM_NAME | Der angegebene Formularname ist ungültig. |
| 1903 | 0x0000076F | ERROR_INVALID_FORM_SIZE | Die angegebene Formulargröße ist ungültig. |
| 1904 | 0x00000770 | ERROR_ALREADY_WAITING | Auf wird das angegebene Druckerhandle bereits gewartet. |
| 1905 | 0x00000771 | ERROR_PRINTER_DELETED | Der angegebene Drucker wurde gelöscht. |
| 1906 | 0x00000772 | ERROR_INVALID_PRINTER_STATE | Der Status des Druckers ist ungültig. |
| 1907 | 0x00000773 | ERROR_PASSWORD_MUST_CHANGE | Das Kennwort des Benutzers muss geändert werden, bevor Sie zum ersten Mal anmelden. |
| 1908 | 0x00000774 | ERROR_DOMAIN_CONTROLLER_NOT_FOUND | Den Domänencontroller konnte für diese Domäne nicht finden werden. |
| 1909 | 0x00000775 | ERROR_ACCOUNT_LOCKED_OUT | Das angesprochene Konto ist momentan gesperrt und kann nicht werden angemeldet. |
| 1910 | 0x00000776 | OR_INVALID_OXID | Der Objekt-Ausführer angegeben wurde nicht gefunden. |
| 1911 | 0x00000777 | OR_INVALID_OID | Das angegebene Objekt wurde nicht gefunden. |
| 1912 | 0x00000778 | OR_INVALID_SET | Die angegebene Objektauswertung wurde nicht gefunden. |
| 1913 | 0x00000779 | RPC_S_SEND_INCOMPLETE | Einige Daten verbleiben in der Anfrage-Puffer gesendet werden. |
| 1914 | 0x0000077A | RPC_S_INVALID_ASYNC_HANDLE | Ungültiger asynchroner remote Prozedur Call-Handle. |
| 1915 | 0x0000077B | RPC_S_INVALID_ASYNC_CALL | Ungültige asynchrone RPC-Call-Handle für diesen Vorgang. |
| 1916 | 0x0000077C | RPC_X_PIPE_CLOSED | Das RPC-Pipe-Objekt wurde bereits geschlossen. |
| 1917 | 0x0000077D | RPC_X_PIPE_DISCIPLINE_ERROR | Der RPC-Aufruf abgeschlossen, bevor alle Pipes verarbeitet wurden. |
| 1918 | 0x0000077E | RPC_X_PIPE_EMPTY | Keine weiteren Daten ist von der RPC-Pipe verfügbar. |
| 1919 | 0x0000077F | ERROR_NO_SITENAME | Kein Sitename steht für diese Maschine. |
| 1920 | 0x00000780 | ERROR_CANT_ACCESS_FILE | Die Datei kann nicht vom System zugegriffen werden. |
| 1921 | 0x00000781 | ERROR_CANT_RESOLVE_FILENAME | Der Name der Datei kann nicht vom System nicht aufgelöst werden. |
| 1922 | 0x00000782 | RPC_S_ENTRY_TYPE_MISMATCH | Der Eintrag ist nicht dem erwarteten Typ. |
| 1923 | 0x00000783 | RPC_S_NOT_ALL_OBJS_EXPORTED | Nicht alle Objekt-UUIDs konnten in den angegebenen Eintrag exportiert werden. |
| 1924 | 0x00000784 | RPC_S_INTERFACE_NOT_EXPORTED | Schnittstelle konnte nicht in den angegebenen Eintrag exportiert werden. |
| 1925 | 0x00000785 | RPC_S_PROFILE_NOT_ADDED | Der angegebene Profileintrag konnte nicht hinzugefügt werden. |
| 1926 | 0x00000786 | RPC_S_PRF_ELT_NOT_ADDED | Das angegebene Profilelement konnte nicht hinzugefügt werden. |
| 1927 | 0x00000787 | RPC_S_PRF_ELT_NOT_REMOVED | Das angegebene Profilelement konnte nicht entfernt werden. |
| 1928 | 0x00000788 | RPC_S_GRP_ELT_NOT_ADDED | Das Gruppenelement konnte nicht hinzugefügt werden. |
| 1929 | 0x00000789 | RPC_S_GRP_ELT_NOT_REMOVED | Das Gruppenelement konnte nicht entfernt werden. |
| 1930 | 0x0000078A | ERROR_KM_DRIVER_BLOCKED | Der Druckertreiber ist nicht kompatibel mit einer Richtlinie auf Ihrem Computer NT 4.0-Treiber blockiert aktiviert. |

| Fehler | | | Beschreibung |
|---------|-------------|--|--|
| Dezimal | Hexadezimal | Name | |
| 2000 | 0x000007D0 | ERROR_INVALID_PIXEL_FORMAT | Das Pixelformat ist ungültig. |
| 2001 | 0x000007D1 | ERROR_BAD_DRIVER | Der angegebene Treiber ist ungültig. |
| 2002 | 0x000007D2 | ERROR_INVALID_WINDOW_STYLE | Das Fenster-Stil oder Class-Attribut ist ungültig für diesen Vorgang. |
| 2003 | 0x000007D3 | ERROR_METAFILE_NOT_SUPPORTED | Der angeforderte Metafilevorgang wird nicht unterstützt. |
| 2004 | 0x000007D4 | ERROR_TRANSFORM_NOT_SUPPORTED | Der Transformation der angeforderte Vorgang wird nicht unterstützt. |
| 2005 | 0x000007D5 | ERROR_CLIPPING_NOT_SUPPORTED | Der angeforderte Clipping-Vorgang wird nicht unterstützt. |
| 2010 | 0x000007DA | ERROR_INVALID_CMM | Das angegebene Farbverwaltungsmodul ist ungültig. |
| 2011 | 0x000007DB | ERROR_INVALID_PROFILE | Das angegebene Farbprofil ist ungültig. |
| 2012 | 0x000007DC | ERROR_TAG_NOT_FOUND | Das angegebene Tag wurde nicht gefunden. |
| 2013 | 0x000007DD | ERROR_TAG_NOT_PRESENT | Ein erforderliche Tag ist nicht vorhanden. |
| 2014 | 0x000007DE | ERROR_DUPLICATE_TAG | Das angegebene Tag ist bereits vorhanden. |
| 2015 | 0x000007DF | ERROR_PROFILE_NOT_ASSOCIATED_WITH_DEVICE | Das angegebene Farbprofil ist nicht mit jedem Gerät verbunden. |
| 2016 | 0x000007E0 | ERROR_PROFILE_NOT_FOUND | Das angegebene Farbprofil wurde nicht gefunden. |
| 2017 | 0x000007E1 | ERROR_INVALID_COLORSPACE | Der angegebenen Farbraum ist ungültig. |
| 2018 | 0x000007E2 | ERROR_ICM_NOT_ENABLED | Farbverwaltung ist nicht aktiviert. |
| 2019 | 0x000007E3 | ERROR_DELETING_ICM_XFORM | Beim Löschen der Farbe-Transformation ist ein Fehler aufgetreten. |
| 2020 | 0x000007E4 | ERROR_INVALID_TRANSFORM | Die angegebene Farbe-Transformation ist ungültig. |
| 2021 | 0x000007E5 | ERROR_COLORSPACE_MISMATCH | Die angegebene Transformation stimmt nicht überein, die Bitmap-Farbraum. |
| 2022 | 0x000007E6 | ERROR_INVALID_COLORINDEX | Der angegebene benannte Farbindex ist nicht im Profil vorhanden. |
| 2108 | 0x0000083C | ERROR_CONNECTED_OTHER_PASSWORD | Die Netzwerkverbindung wurde erfolgreich hergestellt, aber der Benutzer musste aufgefordert, ein Kennwort nicht dem ursprünglich angegebenen übereinstimmte. |
| 2202 | 0x0000089A | ERROR_BAD_USERNAME | Der angegebene Benutzername ist ungültig. |
| 2250 | 0x000008CA | ERROR_NOT_CONNECTED | Diese Netzwerkverbindung ist nicht vorhanden. |
| 2401 | 0x00000961 | ERROR_OPEN_FILES | Diese Netzwerkverbindung hat Dateien öffnen oder anstehende Anforderungen. |
| 2402 | 0x00000962 | ERROR_ACTIVE_CONNECTIONS | Aktive Verbindungen sind noch vorhanden. |
| 2404 | 0x00000964 | ERROR_DEVICE_IN_USE | Das Gerät wird von einem aktiven Prozess verwendet und kann nicht getrennt werden. |
| 2500 | 0x000009C4 | ERROR_PKINIT_FAILURE | Das Kerberos-Protokoll-Fehler während der Überprüfung des KDC-Zertifikats während der Smartcard-Anmeldung. |
| 2501 | 0x000009C5 | ERROR_SMARTCARD_SUBSYSTEM_FAILURE | Das Kerberos-Protokoll-Fehler beim Versuch, das Smartcard-Subsystem zu nutzen. |

| Fehler | | | Beschreibung |
|---------|-------------|---|---|
| Dezimal | Hexadezimal | Name | |
| 3000 | 0x00000BB8 | ERROR_UNKNOWN_PRINT_MONITOR | Der angegebene Druckmonitor ist unbekannt. |
| 3001 | 0x00000BB9 | ERROR_PRINTER_DRIVER_IN_USE | Der angegebene Druckertreiber wird zurzeit verwendet. |
| 3002 | 0x00000BBA | ERROR_SPOOL_FILE_NOT_FOUND | Die Spooldatei wurde nicht gefunden. |
| 3003 | 0x00000BBB | ERROR_SPL_NO_STARTDOC | Ein StartDocPrinter wurde nicht ausgeführt. |
| 3004 | 0x00000BBC | ERROR_SPL_NO_ADDJOB | Ein AddJob-Aufruf wurde nicht ausgestellt. |
| 3005 | 0x00000BBD | ERROR_PRINT_PROCESSOR_ALREADY_INSTALLED | Der angegebene Druckprozessor ist bereits installiert. |
| 3006 | 0x00000BBE | ERROR_PRINT_MONITOR_ALREADY_INSTALLED | Der angegebene Druckmonitor ist bereits installiert. |
| 3007 | 0x00000BBF | ERROR_INVALID_PRINT_MONITOR | Der angegebene Druckmonitor muss nicht die benötigten Funktionen. |
| 3008 | 0x00000BC0 | ERROR_PRINT_MONITOR_IN_USE | Der angegebene Druckmonitor ist momentan in Gebrauch. |
| 3009 | 0x00000BC1 | ERROR_PRINTER_HAS_JOBS_QUEUED | Der angeforderte Vorgang ist nicht zulässig, wenn gibt es Arbeitsplätze, die an den Drucker in eine Warteschlange gestellt. |
| 3010 | 0x00000BC2 | ERROR_SUCCESS_REBOOT_REQUIRED | Der angeforderte Vorgang ist erfolgreich. Änderungen werden nicht wirksam, bis das System neu gestartet wird. |
| 3011 | 0x00000BC3 | ERROR_SUCCESS_RESTART_REQUIRED | Der angeforderte Vorgang ist erfolgreich. Änderungen werden nicht wirksam, bis der Dienst neu gestartet wird. |
| 3012 | 0x00000BC4 | ERROR_PRINTER_NOT_FOUND | Keine Drucker es wurden gefunden. |
| 4000 | 0x00000FA0 | ERROR_WINS_INTERNAL | WINS-Fehler beim Verarbeiten des Befehls. |
| 4001 | 0x00000FA1 | ERROR_CAN_NOT_DEL_LOCAL_WINS | Der lokale WINS können nicht gelöscht werden. |
| 4002 | 0x00000FA2 | ERROR_STATIC_INIT | Die Einfuhr aus der Datei ist fehlgeschlagen. |
| 4003 | 0x00000FA3 | ERROR_INC_BACKUP | Die Sicherung ist fehlgeschlagen. Erfolgte eine vollständige Sicherung vor? |
| 4004 | 0x00000FA4 | ERROR_FULL_BACKUP | Die Sicherung ist fehlgeschlagen. Überprüfen Sie das Verzeichnis, in dem Sie die Datenbank sichern möchten. |
| 4005 | 0x00000FA5 | ERROR_REC_NON_EXISTENT | Der Name ist nicht in der WINS-Datenbank vorhanden. |
| 4006 | 0x00000FA6 | ERROR_RPL_NOT_ALLOWED | Replikation mit einem nicht konfigurierten Partner ist nicht erlaubt. |
| 4100 | 0x00001004 | ERROR_DHCP_ADDRESS_CONFLICT | Der DHCP-Client hat eine IP-Adresse erhalten, die bereits im Netzwerk verwendet wird. Die lokale Schnittstelle wird deaktiviert werden, bis der DHCP-Client eine neue Adresse abrufen kann. |
| 4200 | 0x00001068 | ERROR_WMI_GUID_NOT_FOUND | Die GUID übergeben wurde von einem WMI-Datenprovider nicht als gültig anerkannt. |
| 4201 | 0x00001069 | ERROR_WMI_INSTANCE_NOT_FOUND | Den Namen der Instanz übergeben wurde von einem WMI-Datenprovider nicht als gültig anerkannt. |
| 4202 | 0x0000106A | ERROR_WMI_ITEMID_NOT_FOUND | Die Daten-Element-ID übergeben wurde von einem WMI-Datenprovider nicht als gültig anerkannt. |
| 4203 | 0x0000106B | ERROR_WMI_TRY_AGAIN | Die WMI-Anforderung konnte nicht abgeschlossen werden und sollte wiederholt werden. |
| 4204 | 0x0000106C | ERROR_WMI_DP_NOT_FOUND | Der WMI-Datenanbieter konnte nicht gefunden werden. |
| 4205 | 0x0000106D | ERROR_WMI_UNRESOLVED_INSTANCE_REF | Der WMI-Datenanbieter verweist auf eine Instanz-Gruppe, die nicht registriert wurde. |
| 4206 | 0x0000106E | ERROR_WMI_ALREADY_ENABLED | Der WMI-Datenblock oder die Ereignisbenachrichtigung wurde bereits aktiviert. |
| 4207 | 0x0000106F | ERROR_WMI_GUID_DISCONNECTED | Der WMI-Datenblock ist nicht mehr verfügbar. |
| 4208 | 0x00001070 | ERROR_WMI_SERVER_UNAVAILABLE | Der WMI-Datendienst ist nicht verfügbar. |
| 4209 | 0x00001071 | ERROR_WMI_DP_FAILED | Der WMI-Datenanbieter konnte die Anforderung durchzuführen. |
| 4210 | 0x00001072 | ERROR_WMI_INVALID_MOF | Die WMI-MOF-Informationen sind ungültig. |
| 4211 | 0x00001073 | ERROR_WMI_INVALID_REGINFO | Die WMI-Registrierungsinformationen ist nicht gültig. |

| Fehler | | | Beschreibung |
|---------|-------------|-------------------------------------|--|
| Dezimal | Hexadezimal | Name | |
| 4212 | 0x00001074 | ERROR_WMI_ALREADY_DISABLED | Der WMI-Datenblock oder die Ereignisbenachrichtigung wurde bereits deaktiviert. |
| 4213 | 0x00001075 | ERROR_WMI_READ_ONLY | Der WMI-Objekt oder Daten-Datenblock ist schreibgeschützt. |
| 4214 | 0x00001076 | ERROR_WMI_SET_FAILURE | Der WMI-Objekt oder Daten-Datenblock konnte nicht geändert werden. |
| 4300 | 0x000010CC | ERROR_INVALID_MEDIA | Die Medienkennung stellt kein gültiges Medium dar. |
| 4301 | 0x000010CD | ERROR_INVALID_LIBRARY | Die Bibliothek-ID stellt keine gültige Bibliothek dar. |
| 4302 | 0x000010CE | ERROR_INVALID_MEDIA_POOL | Die Medienkennung Pool stellt keinen gültigen Medienpool dar. |
| 4303 | 0x000010CF | ERROR_DRIVE_MEDIA_MISMATCH | Das Laufwerk und das Medium sind nicht kompatibel, oder in verschiedenen Bibliotheken vorhanden. |
| 4304 | 0x000010D0 | ERROR_MEDIA_OFFLINE | Das Medium muss derzeit in einer Offlinebibliothek vorhanden und online sein, um diesen Vorgang auszuführen. |
| 4305 | 0x000010D1 | ERROR_LIBRARY_OFFLINE | Der Vorgang kann nicht auf einer Offlinebibliothek ausgeführt werden. |
| 4306 | 0x000010D2 | ERROR_EMPTY | Der Bibliothek, das Laufwerk oder Medien-Pool ist leer. |
| 4307 | 0x000010D3 | ERROR_NOT_EMPTY | Bibliothek, das Laufwerk oder der Medienpool muss zum Ausführen dieses Vorgangs leer sein. |
| 4308 | 0x000010D4 | ERROR_MEDIA_UNAVAILABLE | Kein Medium ist derzeit in diesem Medienpool oder Bibliothek. |
| 4309 | 0x000010D5 | ERROR_RESOURCE_DISABLED | Eine Ressource, die für diesen Vorgang erforderlich ist deaktiviert. |
| 4310 | 0x000010D6 | ERROR_INVALID_CLEANER | Die Medienkennung vertritt keinen gültigen Cleaner. |
| 4311 | 0x000010D7 | ERROR_UNABLE_TO_CLEAN | Das Laufwerk kann nicht gesäubert werden oder Reinigung nicht unterstützt. |
| 4312 | 0x000010D8 | ERROR_OBJECT_NOT_FOUND | Die Objekt-ID stellt kein gültiges Objekt dar. |
| 4313 | 0x000010D9 | ERROR_DATABASE_FAILURE | Nicht in der Lage zu lesen oder schreiben in die Datenbank. |
| 4314 | 0x000010DA | ERROR_DATABASE_FULL | Die Datenbank ist voll. |
| 4315 | 0x000010DB | ERROR_MEDIA_INCOMPATIBLE | Das Medium ist nicht kompatibel mit dem Gerät oder Medium-Pool. |
| 4316 | 0x000010DC | ERROR_RESOURCE_NOT_PRESENT | Die Ressource benötigt für diesen Vorgang ist nicht vorhanden. |
| 4317 | 0x000010DD | ERROR_INVALID_OPERATION | Die Vorgangs-ID ist ungültig. |
| 4318 | 0x000010DE | ERROR_MEDIA_NOT_AVAILABLE | Das Medium ist nicht eingebaut oder für den Einsatz bereit. |
| 4319 | 0x000010DF | ERROR_DEVICE_NOT_AVAILABLE | Das Gerät ist nicht für den Einsatz bereit. |
| 4320 | 0x000010E0 | ERROR_REQUEST_REFUSED | Der Operator oder der Administrator hat die Anforderung abgelehnt. |
| 4321 | 0x000010E1 | ERROR_INVALID_DRIVE_OBJECT | Die Laufwerk-ID repräsentiert kein gültiges Laufwerk. |
| 4322 | 0x000010E2 | ERROR_LIBRARY_FULL | Bibliothek ist voll. Kein Steckplatz steht zur Verfügung. |
| 4323 | 0x000010E3 | ERROR_MEDIUM_NOT_ACCESSIBLE | Der Transport kann nicht das Medium zugreifen. |
| 4324 | 0x000010E4 | ERROR_UNABLE_TO_LOAD_MEDIUM | Nicht in der Lage, das Medium in das Laufwerk zu laden. |
| 4325 | 0x000010E5 | ERROR_UNABLE_TO_INVENTORY_DRIVE | Kann nicht zum Abrufen des Status über das Laufwerk. |
| 4326 | 0x000010E6 | ERROR_UNABLE_TO_INVENTORY_SLOT | Kann nicht zum Abrufen des Status über den Steckplatz. |
| 4327 | 0x000010E7 | ERROR_UNABLE_TO_INVENTORY_TRANSPORT | Kann nicht zum Abrufen des Status über den Transport. |
| 4328 | 0x000010E8 | ERROR_TRANSPORT_FULL | Den Transport kann nicht verwenden werden, da es bereits verwendet wird. |
| 4329 | 0x000010E9 | ERROR_CONTROLLING_IEPORT | Kann nicht öffnen oder schließen das Einlegen bzw. Ausgabefach. |

| Fehler | | | Beschreibung |
|---------|-------------|-------------------------------------|--|
| Dezimal | Hexadezimal | Name | |
| 4330 | 0x000010EA | ERROR_UNABLE_TO_EJECT_MOUNTED_MEDIA | Nicht in der Lage, den Datenträger auszuwerfen, weil es im Auto ist. |
| 4331 | 0x000010EB | ERROR_CLEANER_SLOT_SET | Ein Reinigungssteckplatz ist bereits reserviert. |
| 4332 | 0x000010EC | ERROR_CLEANER_SLOT_NOT_SET | Ein Reinigungssteckplatz ist nicht reserviert. |
| 4333 | 0x000010ED | ERROR_CLEANER_CARTRIDGE_SPENT | Die Reinigungskassette hat die maximale Anzahl von Laufwerk Reinigungen durchgeführt. |
| 4334 | 0x000010EE | ERROR_UNEXPECTED_OMID | Unerwartete auf Medium Bezeichner. |
| 4335 | 0x000010EF | ERROR_CANT_DELETE_LAST_ITEM | Das letzte verbleibende Element in dieser Gruppe oder Ressource kann nicht gelöscht werden. |
| 4336 | 0x000010F0 | ERROR_MESSAGE_EXCEEDS_MAX_SIZE | Der jeweiligen Meldung überschreitet die maximale zulässige Größe für diesen Parameter. |
| 4337 | 0x000010F1 | ERROR_VOLUME_CONTAINS_SYS_FILES | Der Band enthält System- oder Auslagerungsdateien. |
| 4338 | 0x000010F2 | ERROR_INDIGENOUS_TYPE | Der Medientyp kann nicht aus dieser Bibliothek entfernt werden, da mindestens ein Laufwerk in der Bibliothek meldet, dass es diesen Medientyp unterstützen kann. |
| 4339 | 0x000010F3 | ERROR_NO_SUPPORTING_DRIVES | Dieses Offlinemedium kann auf diesem System bereitgestellt werden, da keine aktivierten Laufwerke vorhanden sind, die benutzt werden können. |
| 4340 | 0x000010F4 | ERROR_CLEANER_CARTRIDGE_INSTALLED | Eine Reinigungskassette ist vorhanden in der Bandbibliothek. |
| 4350 | 0x000010FE | ERROR_FILE_OFFLINE | Der Remotespeicherdienst konnte nicht die Datei erinnern. |
| 4351 | 0x000010FF | ERROR_REMOTE_STORAGE_NOT_ACTIVE | Der Remotespeicherdienst ist derzeit nicht betriebsbereit. |
| 4352 | 0x00001100 | ERROR_REMOTE_STORAGE_MEDIA_ERROR | Der Remotespeicherdienst ist einen Medienfehler aufgetreten. |
| 4390 | 0x00001126 | ERROR_NOT_A_REPARSE_POINT | Die Datei oder das Verzeichnis ist kein Analysepunkt. |
| 4391 | 0x00001127 | ERROR_REPARSE_ATTRIBUTE_CONFLICT | Das Analysepunkte Punkt-Attribut kann nicht festgelegt werden, weil es mit einem vorhandenen Attribut widerspricht. |
| 4392 | 0x00001128 | ERROR_INVALID_REPARSE_DATA | Die im Analysepunktpuffer vorhandenen Daten ist ungültig. |
| 4393 | 0x00001129 | ERROR_REPARSE_TAG_INVALID | Die Marke vorhanden im Analysepunktpuffer ist ungültig. |
| 4394 | 0x0000112A | ERROR_REPARSE_TAG_MISMATCH | Es besteht eine Diskrepanz zwischen dem in der Anfrage angegebenen Tag und dem Tag in der Analysepunkt vorhanden. |
| 4500 | 0x00001194 | ERROR_VOLUME_NOT_SIS_ENABLED | Single Instance Storage steht nicht auf diesem Volume. |

| Fehler | | | Beschreibung |
|---------|-------------|---|---|
| Dezimal | Hexadezimal | Name | |
| 5001 | 0x00001389 | ERROR_DEPENDENT_RESOURCE_EXISTS | Die Clusterressource kann nicht in eine andere Gruppe verschoben werden, da andere Ressourcen abhängig sind. |
| 5002 | 0x0000138A | ERROR_DEPENDENCY_NOT_FOUND | Die Clusterressourcenabhängigkeit wurde nicht gefunden. |
| 5003 | 0x0000138B | ERROR_DEPENDENCY_ALREADY_EXISTS | Die Clusterressource kann nicht der angegebenen Ressource abhängig gemacht werden, da er bereits angewiesen ist. |
| 5004 | 0x0000138C | ERROR_RESOURCE_NOT_ONLINE | Die Cluster-Ressource ist nicht online. |
| 5005 | 0x0000138D | ERROR_HOST_NODE_NOT_AVAILABLE | Ein Clusterknoten ist nicht verfügbar für diesen Vorgang. |
| 5006 | 0x0000138E | ERROR_RESOURCE_NOT_AVAILABLE | Die Cluster-Ressource ist nicht verfügbar. |
| 5007 | 0x0000138F | ERROR_RESOURCE_NOT_FOUND | Die Clusterressource konnte nicht gefunden werden. |
| 5008 | 0x00001390 | ERROR_SHUTDOWN_CLUSTER | Der Cluster wird heruntergefahren. |
| 5009 | 0x00001391 | ERROR_CANT_EVICT_ACTIVE_NODE | Ein Clusterknoten kann nicht vom Cluster entfernt, es sei denn der Knoten ausfällt. |
| 5010 | 0x00001392 | ERROR_OBJECT_ALREADY_EXISTS | Das Objekt ist bereits vorhanden. |
| 5011 | 0x00001393 | ERROR_OBJECT_IN_LIST | Das Objekt ist bereits in der Liste. |
| 5012 | 0x00001394 | ERROR_GROUP_NOT_AVAILABLE | Die Cluster-Gruppe ist nicht verfügbar für alle neuen Anforderungen. |
| 5013 | 0x00001395 | ERROR_GROUP_NOT_FOUND | Die Cluster-Gruppe konnte nicht gefunden werden. |
| 5014 | 0x00001396 | ERROR_GROUP_NOT_ONLINE | Der Vorgang konnte nicht abgeschlossen werden, da die Clustergruppe nicht online ist. |
| 5015 | 0x00001397 | ERROR_HOST_NODE_NOT_RESOURCE_OWNER | Der Clusterknoten ist nicht der Besitzer der Ressource. |
| 5016 | 0x00001398 | ERROR_HOST_NODE_NOT_GROUP_OWNER | Der Clusterknoten ist nicht der Besitzer der Gruppe. |
| 5017 | 0x00001399 | ERROR_RESMON_CREATE_FAILED | Die Clusterressource konnte in dem angegebenen Ressourcenmonitor nicht erstellt werden. |
| 5018 | 0x0000139A | ERROR_RESMON_ONLINE_FAILED | Die Clusterressource konnte vom Ressourcenmonitor nicht online geschaltet werden. |
| 5019 | 0x0000139B | ERROR_RESOURCE_ONLINE | Der Vorgang konnte nicht abgeschlossen werden, da die Cluster-Ressource online ist. |
| 5020 | 0x0000139C | ERROR_QUORUM_RESOURCE | Die Clusterressource konnte nicht gelöscht oder offline geschaltet werden, da es die Quorumressource ist. |
| 5021 | 0x0000139D | ERROR_NOT_QUORUM_CAPABLE | Der Cluster konnte nicht der angegebenen Ressource eine Quorumressource, da es nicht wird eine Quorumressource kann. |
| 5022 | 0x0000139E | ERROR_CLUSTER_SHUTTING_DOWN | Die Clustersoftware wird heruntergefahren. |
| 5023 | 0x0000139F | ERROR_INVALID_STATE | Die Gruppe oder Ressource ist nicht im richtigen Zustand zum Ausführen des angeforderten Vorgangs. |
| 5024 | 0x000013A0 | ERROR_RESOURCE_PROPERTIES_STORED | Die Eigenschaften wurden gespeichert, aber nicht alle Änderungen werden wirksam, bis das nächste Mal, das die Ressource online geschaltet wird. |
| 5025 | 0x000013A1 | ERROR_NOT_QUORUM_CLASS | Der Cluster konnte nicht der angegebenen Ressource eine Quorumressource, da es nicht zu einer freigegebenen Speicherklasse gehört. |
| 5026 | 0x000013A2 | ERROR_CORE_RESOURCE | Die Clusterressource konnte nicht gelöscht werden, da es sich um eine Kernressource handelt. |
| 5027 | 0x000013A3 | ERROR_QUORUM_RESOURCE_ONLINE_FAILED | Die Quorumressource konnte nicht online geschaltet. |
| 5028 | 0x000013A4 | ERROR_QUORUMLOG_OPEN_FAILED | Die Quorum-Protokolldatei konnten nicht erstellt oder montiert erfolgreich. |
| 5029 | 0x000013A5 | ERROR_CLUSTERLOG_CORRUPT | Die Clusterprotokolldatei ist fehlerhaft. |
| 5030 | 0x000013A6 | ERROR_CLUSTERLOG_RECORD_EXCEEDS_MAXSIZE | Der Datensatz konnte nicht in das Clusterprotokoll geschrieben werden, da sie die maximale Größe überschreitet. |

| Fehler | | | Beschreibung |
|---------|-------------|--|---|
| Dezimal | Hexadezimal | Name | |
| 5031 | 0x000013A7 | ERROR_CLUSTERLOG_EXCEEDS_MAXSIZE | Das Clusterprotokoll überschreitet die maximale Größe. |
| 5032 | 0x000013A8 | ERROR_CLUSTERLOG_CHKPOINT_NOT_FOUND | Kein Prüfpunktdatensatz wurde im Clusterprotokoll gefunden. |
| 5033 | 0x000013A9 | ERROR_CLUSTERLOG_NOT_ENOUGH_SPACE | Der minimal benötigte Speicherplatz für die Protokollierung erforderlich nicht verfügbar ist. |
| 5034 | 0x000013AA | ERROR_QUORUM_OWNER_ALIVE | Der Clusterknoten konnte die Kontrolle über die Quorumressource zu nehmen, da die Ressource einem anderen aktiven Knoten gehört. |
| 5035 | 0x000013AB | ERROR_NETWORK_NOT_AVAILABLE | Ein Cluster-Netzwerk ist nicht verfügbar für diesen Vorgang. |
| 5036 | 0x000013AC | ERROR_NODE_NOT_AVAILABLE | Ein Clusterknoten ist nicht verfügbar für diesen Vorgang. |
| 5037 | 0x000013AD | ERROR_ALL_NODES_NOT_AVAILABLE | Alle Clusterknoten müssen ausgeführt werden, um diesen Vorgang auszuführen. |
| 5038 | 0x000013AE | ERROR_RESOURCE_FAILED | Eine Clusterressource ist fehlgeschlagen. |
| 5039 | 0x000013AF | ERROR_CLUSTER_INVALID_NODE | Der Clusterknoten ist nicht gültig. |
| 5040 | 0x000013B0 | ERROR_CLUSTER_NODE_EXISTS | Der Clusterknoten ist bereits vorhanden. |
| 5041 | 0x000013B1 | ERROR_CLUSTER_JOIN_IN_PROGRESS | Ein Knoten ist dabei, den Cluster beitreten. |
| 5042 | 0x000013B2 | ERROR_CLUSTER_NODE_NOT_FOUND | Der Clusterknoten wurde nicht gefunden. |
| 5043 | 0x000013B3 | ERROR_CLUSTER_LOCAL_NODE_NOT_FOUND | Die Informationen zur Cluster-lokalen Knoten wurde nicht gefunden. |
| 5044 | 0x000013B4 | ERROR_CLUSTER_NETWORK_EXISTS | Das Clusternetzwerk ist bereits vorhanden. |
| 5045 | 0x000013B5 | ERROR_CLUSTER_NETWORK_NOT_FOUND | Das Clusternetzwerk wurde nicht gefunden. |
| 5046 | 0x000013B6 | ERROR_CLUSTER_NETINTERFACE_EXISTS | Die Cluster-Netzwerk-Schnittstelle ist bereits vorhanden. |
| 5047 | 0x000013B7 | ERROR_CLUSTER_NETINTERFACE_NOT_FOUND | Die Cluster-Netzwerk-Schnittstelle wurde nicht gefunden. |
| 5048 | 0x000013B8 | ERROR_CLUSTER_INVALID_REQUEST | Die Cluster-Anforderung gilt nicht für dieses Objekt. |
| 5049 | 0x000013B9 | ERROR_CLUSTER_INVALID_NETWORK_PROVIDER | Die Cluster-Netzwerk-Provider ist nicht gültig. |
| 5050 | 0x000013BA | ERROR_CLUSTER_NODE_DOWN | Der Clusterknoten ist unten. |
| 5051 | 0x000013BB | ERROR_CLUSTER_NODE_UNREACHABLE | Der Clusterknoten ist nicht erreichbar. |
| 5052 | 0x000013BC | ERROR_CLUSTER_NODE_NOT_MEMBER | Der Clusterknoten ist kein Mitglied des Clusters. |
| 5053 | 0x000013BD | ERROR_CLUSTER_JOIN_NOT_IN_PROGRESS | Eine Cluster-Join-Operation wird nicht ausgeführt. |
| 5054 | 0x000013BE | ERROR_CLUSTER_INVALID_NETWORK | Das Clusternetzwerk ist nicht gültig. |
| 5056 | 0x000013C0 | ERROR_CLUSTER_NODE_UP | Der Clusterknoten ist oben. |
| 5057 | 0x000013C1 | ERROR_CLUSTER_IPADDR_IN_USE | Die IP-Adresse des Clusters wird bereits verwendet. |
| 5058 | 0x000013C2 | ERROR_CLUSTER_NODE_NOT_PAUSED | Der Clusterknoten wurde nicht angehalten. |
| 5059 | 0x000013C3 | ERROR_CLUSTER_NO_SECURITY_CONTEXT | Keine Cluster-Sicherheitskontext steht zur Verfügung. |
| 5060 | 0x000013C4 | ERROR_CLUSTER_NETWORK_NOT_INTERNAL | Das Clusternetzwerk ist nicht für die interne Clusterkommunikation konfiguriert. |
| 5061 | 0x000013C5 | ERROR_CLUSTER_NODE_ALREADY_UP | Der Clusterknoten ist bereits oben. |
| 5062 | 0x000013C6 | ERROR_CLUSTER_NODE_ALREADY_DOWN | Der Clusterknoten ist bereits nach unten. |
| 5063 | 0x000013C7 | ERROR_CLUSTER_NETWORK_ALREADY_ONLINE | Das Clusternetzwerk ist bereits online. |
| 5064 | 0x000013C8 | ERROR_CLUSTER_NETWORK_ALREADY_OFFLINE | Das Clusternetzwerk ist bereits offline. |
| 5065 | 0x000013C9 | ERROR_CLUSTER_NODE_ALREADY_MEMBER | Der Clusterknoten ist bereits Mitglied des Clusters. |
| 5066 | 0x000013CA | ERROR_CLUSTER_LAST_INTERNAL_NETWORK | Das Clusternetzwerk ist das einzige, das für die interne Clusterkommunikation zwischen mindestens zwei aktiven Clusterknoten konfiguriert. Die interne Kommunikationsfähigkeit kann aus dem Netzwerk entfernt werden. |
| 5067 | 0x000013CB | ERROR_CLUSTER_NETWORK_HAS_DEPENDENTS | Einem oder mehreren Clusterressourcen abhängen im Netzwerk, Service für Kunden. Die Möglichkeit des Client-Zugriff kann nicht aus dem Netzwerk entfernt werden. |

| Fehler | | | Beschreibung |
|---------|-------------|--|--|
| Dezimal | Hexadezimal | Name | |
| 5068 | 0x000013CC | ERROR_INVALID_OPERATION_ON_QUORUM | Dieser Vorgang kann nicht werden ausgeführt auf die Cluster-Ressource, da es die Quorumressource. Sie können nicht die Quorumressource offline zu bringen oder ändern die Liste der möglichen Besitzer. |
| 5069 | 0x000013CD | ERROR_DEPENDENCY_NOT_ALLOWED | Die Cluster-Quorum-Ressource ist nicht zulässig, keine Abhängigkeiten haben. |
| 5070 | 0x000013CE | ERROR_CLUSTER_NODE_PAUSED | Der Clusterknoten ist angehalten. |
| 5071 | 0x000013CF | ERROR_NODE_CANT_HOST_RESOURCE | Die Clusterressource konnte nicht online geschaltet werden. Der Besitzerknoten kann diese Ressource nicht ausführen. |
| 5072 | 0x000013D0 | ERROR_CLUSTER_NODE_NOT_READY | Der Clusterknoten ist nicht bereit sind, die angeforderte Operation auszuführen. |
| 5073 | 0x000013D1 | ERROR_CLUSTER_NODE_SHUTTING_DOWN | Der Clusterknoten wird heruntergefahren. |
| 5074 | 0x000013D2 | ERROR_CLUSTER_JOIN_ABORTED | Der Cluster-Join-Vorgang wurde abgebrochen. |
| 5075 | 0x000013D3 | ERROR_CLUSTER_INCOMPATIBLE_VERSIONS | Die Cluster-Join-Operation scheiterte inkompatible Softwareversionen zwischen beitretenden Knotens und seiner Auftraggeber. |
| 5076 | 0x000013D4 | ERROR_CLUSTER_MAXNUM_OF_RESOURCES_EXCEEDED | Diese Ressource kann nicht erstellt werden, da der Cluster den Grenzwert für die Anzahl der Ressourcen erreicht hat, die sie überwachen kann. |
| 5077 | 0x000013D5 | ERROR_CLUSTER_SYSTEM_CONFIG_CHANGE | Die Konfiguration des Systems während des Cluster beitreten oder Form verändert. Der Join oder Formular-Vorgang wurde abgebrochen. |
| 5078 | 0x000013D6 | ERROR_CLUSTER_RESOURCE_TYPE_NOT_FOUND | Der angegebene Ressourcentyp wurde nicht gefunden. |
| 5079 | 0x000013D7 | ERROR_CLUSTER_RESTYPE_NOT_SUPPORTED | Der angegebene Knoten wird eine Ressource dieses Typs nicht unterstützt. Dies kann durch Versionsinkonsistenzen oder durch das Fehlen der Ressourcen-DLL auf diesem Knoten sein. |
| 5080 | 0x000013D8 | ERROR_CLUSTER_RESNAME_NOT_FOUND | Der angegebene Ressourcennamen wird von dieser Ressourcen-DLL unterstützt. Dies ist möglicherweise aufgrund einer schlechten (oder geänderte) Nachnamen auf die Ressourcen-DLL. |
| 5081 | 0x000013D9 | ERROR_CLUSTER_NO_RPC_PACKAGES_REGISTERED | Mit dem RPC-Server konnte kein Authentifizierungspaket registriert werden. |
| 5082 | 0x000013DA | ERROR_CLUSTER_OWNER_NOT_IN_PREFLIST | Sie können nicht die Gruppe nicht online schalten, weil der Besitzer der Gruppe nicht in der Liste der bevorzugten für die Gruppe. Um den Besitzerknoten für die Gruppe zu ändern, verschieben Sie die Gruppe. |
| 5083 | 0x000013DB | ERROR_CLUSTER_DATABASE_SEQMISMATCH | Der Beitrittsvorgang ist fehlgeschlagen, da die Sequenznummer der Cluster-Datenbank geändert hat oder nicht kompatibel mit dem Spind-Knoten ist. Dies kann während einer Verknüpfungoperation passieren, wenn die Clusterdatenbank während der Verknüpfung geändert wurde. |
| 5084 | 0x000013DC | ERROR_RESMON_INVALID_STATE | Der Ressourcenmonitor ist nicht möglich, die Fail-Operation durchgeführt werden, während die Ressource im aktuellen Zustand ist. Dies kann passieren, wenn die Ressource in einem wartenden Zustand befindet. |
| 5085 | 0x000013DD | ERROR_CLUSTER_GUM_NOT_LOCKER | Ein nicht Locker-Code hat eine Anforderung, die Sperre für eine globale Aktualisierungen zu reservieren. |
| 5086 | 0x000013DE | ERROR_QUORUM_DISK_NOT_FOUND | Der Quorumdatenträger konnte vom Clusterdienst nicht gefunden werden. |
| 5087 | 0x000013DF | ERROR_DATABASE_BACKUP_CORRUPT | Die Sicherung Cluster-Datenbank ist möglicherweise beschädigt. |
| 5088 | 0x000013E0 | ERROR_CLUSTER_NODE_ALREADY_HAS_DFS_ROOT | Ein DFS-Stamm existiert bereits in diesem Clusterknoten. |
| 5089 | 0x000013E1 | ERROR_RESOURCE_PROPERTY_UNCHANGEABLE | Ein Versuch, eine Ressourceneigenschaft zu ändern ist fehlgeschlagen, da der Konflikt mit einer anderen vorhandenen Eigenschaft. |

| Fehler | | | Beschreibung |
|---------|-------------|---|---|
| Dezimal | Hexadezimal | Name | |
| 5890 | 0x00001702 | ERROR_CLUSTER_MEMBERSHIP_INVALID_STATE | Ein Vorgang bezog, die inkompatibel mit dem aktuellen Status der Mitgliedschaft des Knotens ist. |
| 5891 | 0x00001703 | ERROR_CLUSTER_QUORUMLOG_NOT_FOUND | Die Quorumressource enthält nicht die Quorum-Protokolldatei. |
| 5892 | 0x00001704 | ERROR_CLUSTER_MEMBERSHIP_HALT | Das Mitgliedschaftsmodul angefordert Herunterfahren des Clusterdienstes auf diesem Knoten. |
| 5893 | 0x00001705 | ERROR_CLUSTER_INSTANCE_ID_MISMATCH | Der Beitrittsvorgang ist fehlgeschlagen, da die Cluster-Instanz-ID des beitretenden Knotens nicht mit die Cluster-Instanz-ID des Sponsorknoten übereinstimmt. |
| 5894 | 0x00001706 | ERROR_CLUSTER_NETWORK_NOT_FOUND_FOR_IP | Eine passende Netzwerk für die angegebene IP-Adresse konnte nicht gefunden werden. Bitte geben Sie eine Subnetzmaske und ein Clusternetzwerk. |
| 5895 | 0x00001707 | ERROR_CLUSTER_PROPERTY_DATA_TYPE_MISMATCH | Der eigentliche Datentyp der Eigenschaft entsprach nicht den erwarteten Datentyp der Eigenschaft. |
| 5896 | 0x00001708 | ERROR_CLUSTER_EVICT_WITHOUT_CLEANUP | Der Clusterknoten wurde erfolgreich vom Cluster entfernt. Der Knoten wurde nicht bereinigt, da es die entfernen-Cleanup-Funktionalität nicht unterstützt. |

| Fehler | | | Beschreibung |
|---------|-------------|--------------------------------------|---|
| Dezimal | Hexadezimal | Name | |
| 6000 | 0x00001770 | ERROR_ENCRYPTION_FAILED | Die angegebene Datei konnte nicht verschlüsselt werden. |
| 6001 | 0x00001771 | ERROR_DECRYPTION_FAILED | Die angegebene Datei konnte nicht entschlüsselt werden. |
| 6002 | 0x00001772 | ERROR_FILE_ENCRYPTED | Die angegebene Datei ist verschlüsselt und der Benutzer verfügt nicht über die Fähigkeit zur Entschlüsselung. |
| 6003 | 0x00001773 | ERROR_NO_RECOVERY_POLICY | Es gibt keine gültige Verschlüsselung Wiederherstellungsrichtlinie für dieses System. |
| 6004 | 0x00001774 | ERROR_NO_EFS | Der erforderliche Verschlüsselung-Treiber ist nicht für dieses System geladen. |
| 6005 | 0x00001775 | ERROR_WRONG_EFS | Die Datei wurde mit einem Treiber andere Verschlüsselung verschlüsselt, als derzeit geladen wird. |
| 6006 | 0x00001776 | ERROR_NO_USER_KEYS | Es gibt keine EFS-Schlüssel für den Benutzer definiert. |
| 6007 | 0x00001777 | ERROR_FILE_NOT_ENCRYPTED | Die angegebene Datei ist nicht verschlüsselt. |
| 6008 | 0x00001778 | ERROR_NOT_EXPORT_FORMAT | Die angegebene Datei ist nicht in das definierte EFS-Exportformat. |
| 6009 | 0x00001779 | ERROR_FILE_READ_ONLY | Die angegebene Datei ist schreibgeschützt. |
| 6010 | 0x0000177A | ERROR_DIR_EFS_DISALLOWED | Das Verzeichnis wurde für die Verschlüsselung deaktiviert. |
| 6011 | 0x0000177B | ERROR_EFS_SERVER_NOT_TRUSTED | Der Server ist nicht für RAS Verschlüsselungsvorgang vertraut. |
| 6012 | 0x0000177C | ERROR_BAD_RECOVERY_POLICY | Wiederherstellungsrichtlinie für dieses System enthält ungültiges Wiederherstellungszertifikat. |
| 6013 | 0x0000177D | ERROR_EFS_ALG_BLOB_TOO_BIG | Auf die Quelldatei verwendete Verschlüsselungsalgorithmus benötigt einen größeren zentralen Puffer als diejenige auf die Zieldatei. |
| 6014 | 0x0000177E | ERROR_VOLUME_NOT_SUPPORT_EFS | Die Datenträgerpartition unterstützt keine Dateiverschlüsselung. |
| 6118 | 0x000017E6 | ERROR_NO_BROWSER_SERVERS_FOUND | Die Liste der Server für diese Arbeitsgruppe ist derzeit nicht verfügbar. |
| 6200 | 0x00001838 | SCHED_E_SERVICE_NOT_LOCALSYSTEM | Der Taskplannerdienst muss konfiguriert werden, um in das System-Konto ordnungsgemäß auszuführen. Einzelne Aufgaben können so konfiguriert werden, in anderen Konten ausgeführt werden. |
| 7001 | 0x00001B59 | ERROR_CTX_WINSTATION_NAME_INVALID | Der angegebene Sitzungsname ist ungültig. |
| 7002 | 0x00001B5A | ERROR_CTX_INVALID_PD | Der angegebene Protokolltreiber ist ungültig. |
| 7003 | 0x00001B5B | ERROR_CTX_PD_NOT_FOUND | Der angegebene Protokolltreiber wurde im Systempfad nicht gefunden. |
| 7004 | 0x00001B5C | ERROR_CTX_WD_NOT_FOUND | Der angegebene Anschluss-Treiber wurde im Systempfad nicht gefunden. |
| 7005 | 0x00001B5D | ERROR_CTX_CANNOT_MAKE_EVENTLOG_ENTRY | Ein Registrierungsschlüssel für die Ereignisprotokollierung konnte nicht für diese Sitzung erstellt werden. |
| 7006 | 0x00001B5E | ERROR_CTX_SERVICE_NAME_COLLISION | Ein Dienst mit dem gleichen Namen ist bereits auf dem System vorhanden. |
| 7007 | 0x00001B5F | ERROR_CTX_CLOSE_PENDING | Ein Schließvorgang steht für die Sitzung. |
| 7008 | 0x00001B60 | ERROR_CTX_NO_OUTBUF | Es sind keine freien Ausgabepuffer verfügbar. |
| 7009 | 0x00001B61 | ERROR_CTX_MODEM_INF_NOT_FOUND | Das MODEM.INF-Datei wurde nicht gefunden. |
| 7010 | 0x00001B62 | ERROR_CTX_INVALID_MODEMNAME | Der Modemname wurde in MODEM.INF nicht gefunden. |
| 7011 | 0x00001B63 | ERROR_CTX_MODEM_RESPONSE_ERROR | Das Modem hat den Befehl, die ihm nicht angenommen. Stellen Sie sicher, dass der Modemname des konfigurierten angeschlossenen Modems entspricht. |
| 7012 | 0x00001B64 | ERROR_CTX_MODEM_RESPONSE_TIMEOUT | Das Modem antwortet nicht auf den Befehl gesendet. Überprüfen Sie, ob das Modem richtig angeschlossen und eingeschaltet. |
| 7013 | 0x00001B65 | ERROR_CTX_MODEM_RESPONSE_NO_CARRIER | Erkennung des Trägersignals ist fehlgeschlagen oder Carrier wurde gelöscht wegen um zu trennen. |

| Fehler | | | Beschreibung |
|---------|-------------|--------------------------------------|--|
| Dezimal | Hexadezimal | Name | |
| 7014 | 0x00001B66 | ERROR_CTX_MODEM_RESPONSE_NO_DIALTONE | Freizeichen nicht innerhalb der erforderlichen Zeit erkannt. Stellen Sie sicher, dass das Telefonkabel richtig angeschlossen und funktionsfähig ist. |
| 7015 | 0x00001B67 | ERROR_CTX_MODEM_RESPONSE_BUSY | Busy Signal zur remote-Site auf Rückruf erkannt. |
| 7016 | 0x00001B68 | ERROR_CTX_MODEM_RESPONSE_VOICE | An remote-Standort beim Rückruf wurde ein Sprachsignal. |
| 7017 | 0x00001B69 | ERROR_CTX_TD_ERROR | Transport-Treiber-Fehler |
| 7022 | 0x00001B6E | ERROR_CTX_WINSTATION_NOT_FOUND | Die angegebene Sitzung wurde nicht gefunden. |
| 7023 | 0x00001B6F | ERROR_CTX_WINSTATION_ALREADY_EXISTS | Der angegebene Sitzungsname wird bereits verwendet. |
| 7024 | 0x00001B70 | ERROR_CTX_WINSTATION_BUSY | Der angeforderte Vorgang kann nicht werden abgeschlossen, da die Terminalverbindung momentan Verarbeitung einer verbinden, trennen, zurücksetzen oder Löschvorgang. |
| 7025 | 0x00001B71 | ERROR_CTX_BAD_VIDEO_MODE | Ein Versuch wurde unternommen, zum Herstellen einer Verbindung mit einer Sitzungs, deren Videomodus vom aktuellen Client nicht unterstützt wird. |
| 7035 | 0x00001B7B | ERROR_CTX_GRAPHICS_INVALID | Die Anwendung hat versucht, die DOS-Grafikmodus zu aktivieren. DOS-Grafikmodus wird nicht unterstützt. |
| 7037 | 0x00001B7D | ERROR_CTX_LOGON_DISABLED | Ihr Privileg der interaktiven Anmeldung wurde deaktiviert. Bitte kontaktieren Sie Ihren Administrator. |
| 7038 | 0x00001B7E | ERROR_CTX_NOT_CONSOLE | Der angeforderte Vorgang kann nur an der Systemkonsole erfolgen. Dies ist meist das Ergebnis von einem Treiber oder System DLL erfordert direkten Konsolenzugriff. |
| 7040 | 0x00001B80 | ERROR_CTX_CLIENT_QUERY_TIMEOUT | Der Client an den Server reagiert herstellen Nachricht. |
| 7041 | 0x00001B81 | ERROR_CTX_CONSOLE_DISCONNECT | Trennen der Konsolensitzung wird nicht unterstützt. |
| 7042 | 0x00001B82 | ERROR_CTX_CONSOLE_CONNECT | Wiederherstellen einer getrennten Sitzungs auf der Konsole wird nicht unterstützt. |
| 7044 | 0x00001B84 | ERROR_CTX_SHADOW_DENIED | Die Anforderung zur Remoteüberwachung einer anderen Sitzung wurde verweigert. |
| 7045 | 0x00001B85 | ERROR_CTX_WINSTATION_ACCESS_DENIED | Der angeforderte Sitzungszugriff verweigert. |
| 7049 | 0x00001B89 | ERROR_CTX_INVALID_WD | Der angegebene Anschluss Treiber ist ungültig. |
| 7050 | 0x00001B8A | ERROR_CTX_SHADOW_INVALID | Die angeforderte Sitzung kann nicht remote gesteuert werden. Dies kann sein, weil die Sitzung getrennt wird oder verfügt derzeit nicht über ein Benutzer angemeldet. |
| 7051 | 0x00001B8B | ERROR_CTX_SHADOW_DISABLED | Die angeforderte Sitzung ist nicht für die Fernbedienung zulassen konfiguriert. |
| 7052 | 0x00001B8C | ERROR_CTX_CLIENT_LICENSE_IN_USE | Ihre Anfrage zum Herstellen einer Verbindung mit diesem Terminalserver ist abgelehnt worden. Ihre Lizenznummer für Terminal Server-Client ist derzeit von einem anderen Benutzer verwendet wird. Rufen Sie bitte Ihren Systemadministrator, um eine eindeutige Lizenznummer zu erhalten. |
| 7053 | 0x00001B8D | ERROR_CTX_CLIENT_LICENSE_NOT_SET | Ihre Anfrage zum Herstellen einer Verbindung mit diesem Terminalserver ist abgelehnt worden. Ihre Lizenznummer für Terminal Server-Client wurde nicht für diese Kopie des Terminal Server-Clients eingegeben. Bitte kontaktieren Sie Ihren Systemadministrator. |
| 7054 | 0x00001B8E | ERROR_CTX_LICENSE_NOT_AVAILABLE | Das System hat seine lizenzierte Anmelde-limit erreicht. Bitte versuchen Sie es später erneut. |
| 7055 | 0x00001B8F | ERROR_CTX_LICENSE_CLIENT_INVALID | Der Client, den Sie verwenden ist nicht lizenziert, um dieses System zu verwenden. Die Anmeldeanforderung wurde verweigert. |
| 7056 | 0x00001B90 | ERROR_CTX_LICENSE_EXPIRED | Die System-Lizenz ist abgelaufen. Die Anmeldeanforderung wurde verweigert. |
| 7057 | 0x00001B91 | ERROR_CTX_SHADOW_NOT_RUNNING | Fernbedienung konnte nicht abgebrochen werden, weil die angegebene Sitzung nicht gerade aus der Ferne gesteuert wird. |

| Fehler | | | Beschreibung |
|---------|-------------|--------------------------------------|---|
| Dezimal | Hexadezimal | Name | |
| 8001 | 0x00001F41 | FRS_ERR_INVALID_API_SEQUENCE | Die Datei Replikation Dienst-API wurde falsch aufgerufen. |
| 8002 | 0x00001F42 | FRS_ERR_STARTING_SERVICE | Der Dateireplikationsdienst kann nicht gestartet werden. |
| 8003 | 0x00001F43 | FRS_ERR_STOPPING_SERVICE | Der Dateireplikationsdienst kann nicht beendet werden. |
| 8004 | 0x00001F44 | FRS_ERR_INTERNAL_API | Die Datei Replikation Dienst-API wurde die Anforderung beendet. Das Ereignisprotokoll möglicherweise weitere Informationen. |
| 8005 | 0x00001F45 | FRS_ERR_INTERNAL | Der Dateireplikationsdienst beendet die Anforderung. Das Ereignisprotokoll möglicherweise weitere Informationen. |
| 8006 | 0x00001F46 | FRS_ERR_SERVICE_COMM | Der Dateireplikationsdienst kann nicht kontaktiert werden. Das Ereignisprotokoll möglicherweise weitere Informationen. |
| 8007 | 0x00001F47 | FRS_ERR_INSUFFICIENT_PRIV | Der Dateireplikationsdienst kann nicht die Anforderung zu erfüllen, weil der Benutzer nicht über ausreichende Berechtigungen verfügt. Das Ereignisprotokoll möglicherweise weitere Informationen. |
| 8008 | 0x00001F48 | FRS_ERR_AUTHENTICATION | Der Dateireplikationsdienst kann nicht der Anforderung erfüllen, weil authentifizierte RPC nicht verfügbar ist. Das Ereignisprotokoll möglicherweise weitere Informationen. |
| 8009 | 0x00001F49 | FRS_ERR_PARENT_INSUFFICIENT_PRIV | Der Dateireplikationsdienst kann nicht die Anforderung zu erfüllen, weil der Benutzer nicht über ausreichende Berechtigungen auf dem Domänencontroller verfügt. Das Ereignisprotokoll möglicherweise weitere Informationen. |
| 8010 | 0x00001F4A | FRS_ERR_PARENT_AUTHENTICATION | Der Dateireplikationsdienst kann nicht der Anforderung erfüllen, weil authentifizierte RPC nicht verfügbar auf dem Domänencontroller ist. Das Ereignisprotokoll möglicherweise weitere Informationen. |
| 8011 | 0x00001F4B | FRS_ERR_CHILD_TO_PARENT_COMM | Der Dateireplikationsdienst kann nicht mit der Dateireplikationsdienst auf dem Domänencontroller kommunizieren. Das Ereignisprotokoll möglicherweise weitere Informationen. |
| 8012 | 0x00001F4C | FRS_ERR_PARENT_TO_CHILD_COMM | Der Dateireplikationsdienst des Domänencontrollers kann nicht mit der Dateireplikationsdienst auf diesem Computer kommunizieren. Das Ereignisprotokoll möglicherweise weitere Informationen. |
| 8013 | 0x00001F4D | FRS_ERR_SYSVOL_POPULATE | Der Dateireplikationsdienst kann wegen eines internen Fehlers nicht das Systemvolumen ausfüllen. Das Ereignisprotokoll möglicherweise weitere Informationen. |
| 8014 | 0x00001F4E | FRS_ERR_SYSVOL_POPULATE_TIMEOUT | Der Dateireplikationsdienst kann wegen einer internen Zeitüberschreitung nicht das Systemvolumen ausfüllen. Das Ereignisprotokoll möglicherweise weitere Informationen. |
| 8015 | 0x00001F4F | FRS_ERR_SYSVOL_IS_BUSY | Der Dateireplikationsdienst kann die Anforderung nicht verarbeiten. Das Systemvolumen ist mit einer vorherigen Anforderung ausgelastet. |
| 8016 | 0x00001F50 | FRS_ERR_SYSVOL_DEMOTE | Der Dateireplikationsdienst kann nicht beendet die Replikation von Systemdatenträger wegen eines internen Fehlers. Das Ereignisprotokoll möglicherweise weitere Informationen. |
| 8017 | 0x00001F51 | FRS_ERR_INVALID_SERVICE_PARAMETER | Der Dateireplikationsdienst hat einen ungültigen Parameter festgestellt. |
| 8200 | 0x00002008 | ERROR_DS_NOT_INSTALLED | Fehler beim Installieren des Verzeichnisdiensts. Weitere Informationen finden Sie im Ereignisprotokoll. |
| 8201 | 0x00002009 | ERROR_DS_MEMBERSHIP_EVALUATED_LOCALY | Der Verzeichnisdienst ausgewertet Gruppenmitgliedschaften lokal. |
| 8202 | 0x0000200A | ERROR_DS_NO_ATTRIBUTE_OR_VALUE | Die angegebene Verzeichnisdienstattribut oder der Wert ist nicht vorhanden. |

| Fehler | | | Beschreibung |
|---------|-------------|-------------------------------------|---|
| Dezimal | Hexadezimal | Name | |
| 8203 | 0x0000200B | ERROR_DS_INVALID_ATTRIBUTE_SYNTAX | Die Attributsyntax, die an den Verzeichnisdienst ist ungültig. |
| 8204 | 0x0000200C | ERROR_DS_ATTRIBUTE_TYPE_UNDEFINED | Der Typ des Attributs angegeben, um den Verzeichnisdienst ist nicht definiert. |
| 8205 | 0x0000200D | ERROR_DS_ATTRIBUTE_OR_VALUE_EXISTS | Das angegebene Verzeichnisdienstattribut oder Wert ist bereits vorhanden. |
| 8206 | 0x0000200E | ERROR_DS_BUSY | Der Verzeichnisdienst ist ausgelastet. |
| 8207 | 0x0000200F | ERROR_DS_UNAVAILABLE | Der Verzeichnisdienst ist nicht verfügbar. |
| 8208 | 0x00002010 | ERROR_DS_NO_RIDS_ALLOCATED | Der Verzeichnisdienst konnte einen relativen Bezeichner zuweisen. |
| 8209 | 0x00002011 | ERROR_DS_NO_MORE_RIDS | Der Verzeichnisdienst hat den Pool der relative IDs erschöpft. |
| 8210 | 0x00002012 | ERROR_DS_INCORRECT_ROLE_OWNER | Der angeforderte Vorgang konnte nicht ausgeführt werden, da der Verzeichnisdienst nicht den Master für diese Art von Vorgang ist. |
| 8211 | 0x00002013 | ERROR_DS_RIDMGR_INIT_ERROR | Der Verzeichnisdienst konnte das Teilsystem, das relative Identifikatoren zuweist, zu initialisieren. |
| 8212 | 0x00002014 | ERROR_DS_OBJ_CLASS_VIOLATION | Der angeforderte Vorgang hat eine oder mehrere Einschränkungen mit der Klasse des Objekts nicht erfüllt. |
| 8213 | 0x00002015 | ERROR_DS_CANT_ON_NON_LEAF | Der Verzeichnisdienst kann den angeforderten Vorgang nur an einem Endknotenobjekt durchführen. |
| 8214 | 0x00002016 | ERROR_DS_CANT_ON_RDN | Der Verzeichnisdienst kann nicht den angeforderten Vorgang an das RDN-Attribut eines Objekts durchführen. |
| 8215 | 0x00002017 | ERROR_DS_CANT_MOD_OBJ_CLASS | Der Verzeichnisdienst erkennt, einen Versuch, die Objektklasse eines Objekts zu ändern. |
| 8216 | 0x00002018 | ERROR_DS_CROSS_DOM_MOVE_ERROR | Der angeforderte domänenübergreifende Verschiebevorgang konnte nicht ausgeführt werden. |
| 8217 | 0x00002019 | ERROR_DS_GC_NOT_AVAILABLE | Nicht in der Lage, den globalen Katalogserver zu kontaktieren. |
| 8218 | 0x0000201A | ERROR_SHARED_POLICY | Das Richtlinienobjekt ist freigegeben und kann nur im Stammverzeichnis geändert werden. |
| 8219 | 0x0000201B | ERROR_POLICY_OBJECT_NOT_FOUND | Das Richtlinienobjekt ist nicht vorhanden. |
| 8220 | 0x0000201C | ERROR_POLICY_ONLY_IN_DS | Die angeforderten Richtlinieninformationen ist nur im Verzeichnisdienst. |
| 8221 | 0x0000201D | ERROR_PROMOTION_ACTIVE | Eine Heraufstufung von Domänencontrollern ist derzeit aktiv. |
| 8222 | 0x0000201E | ERROR_NO_PROMOTION_ACTIVE | Eine Heraufstufung von Domänencontrollern ist derzeit nicht aktiv |
| 8224 | 0x00002020 | ERROR_DS_OPERATIONS_ERROR | Operationen Fehler. |
| 8225 | 0x00002021 | ERROR_DS_PROTOCOL_ERROR | Ein Protokollfehler ist aufgetreten. |
| 8226 | 0x00002022 | ERROR_DS_TIMELIMIT_EXCEEDED | Das Zeitlimit für diese Anforderung wurde überschritten. |
| 8227 | 0x00002023 | ERROR_DS_SIZELIMIT_EXCEEDED | Das Größenlimit für diese Anforderung wurde überschritten. |
| 8228 | 0x00002024 | ERROR_DS_ADMIN_LIMIT_EXCEEDED | Die administrative Grenze für diese Anforderung wurde überschritten. |
| 8229 | 0x00002025 | ERROR_DS_COMPARE_FALSE | Die Vergleich-Antwort war falsch. |
| 8230 | 0x00002026 | ERROR_DS_COMPARE_TRUE | Die Vergleich-Antwort galt. |
| 8231 | 0x00002027 | ERROR_DS_AUTH_METHOD_NOT_SUPPORTED | Die angeforderte Authentifizierungsmethode wird vom Server nicht unterstützt. |
| 8232 | 0x00002028 | ERROR_DS_STRONG_AUTH_REQUIRED | Eine sicherere Authentifizierungsmethode wird für diesen Server benötigt. |
| 8233 | 0x00002029 | ERROR_DS_INAPPROPRIATE_AUTH | Unangemessene Authentifizierung. |
| 8234 | 0x0000202A | ERROR_DS_AUTH_UNKNOWN | Der Authentifizierungsmechanismus ist unbekannt. |
| 8235 | 0x0000202B | ERROR_DS_REFERRAL | Ein Verweis wurde vom Server zurückgegeben. |
| 8236 | 0x0000202C | ERROR_DS_UNAVAILABLE_CRIT_EXTENSION | Der Server unterstützt nicht die angeforderte kritische Erweiterung. |
| 8237 | 0x0000202D | ERROR_DS_CONFIDENTIALITY_REQUIRED | Diese Anforderung erfordert eine sichere Verbindung. |

| Fehler | | | Beschreibung |
|---------|-------------|-----------------------------------|--|
| Dezimal | Hexadezimal | Name | |
| 8238 | 0x0000202E | ERROR_DS_INAPPROPRIATE_MATCHING | Unangemessene Abgleich verwendet werden. |
| 8239 | 0x0000202F | ERROR_DS_CONSTRAINT_VIOLATION | Eine Beschränkungsverletzung aufgetreten. |
| 8240 | 0x00002030 | ERROR_DS_NO_SUCH_OBJECT | Es gibt kein solches Objekt auf dem Server. |
| 8241 | 0x00002031 | ERROR_DS_ALIAS_PROBLEM | Es gibt eine Alias-Problem. |
| 8242 | 0x00002032 | ERROR_DS_INVALID_DN_SYNTAX | Eine ungültige dn-Syntax es wurde angegeben. |
| 8243 | 0x00002033 | ERROR_DS_IS_LEAF | Das Objekt ist ein Leaf-Objekt. |
| 8244 | 0x00002034 | ERROR_DS_ALIAS_DEREF_PROBLEM | Es ist ein Alias Dereferenzierung Problem. |
| 8245 | 0x00002035 | ERROR_DS_UNWILLING_TO_PERFORM | Der Server ist nicht bereit, die Anforderung zu verarbeiten. |
| 8246 | 0x00002036 | ERROR_DS_LOOP_DETECT | Eine Schleife wurde gefunden. |
| 8247 | 0x00002037 | ERROR_DS_NAMING_VIOLATION | Es ist eine Benennung Verletzung. |
| 8248 | 0x00002038 | ERROR_DS_OBJECT_RESULTS_TOO_LARGE | Die Ergebnismenge ist zu groß. |
| 8249 | 0x00002039 | ERROR_DS_AFFECTS_MULTIPLE_DSAS | Der Vorgang beeinflusst mehrere DSAs |
| 8250 | 0x0000203A | ERROR_DS_SERVER_DOWN | Der Server ist nicht funktionstüchtig. |
| 8251 | 0x0000203B | ERROR_DS_LOCAL_ERROR | Ein lokaler Fehler ist aufgetreten. |
| 8252 | 0x0000203C | ERROR_DS_ENCODING_ERROR | Ein Codierung Fehler aufgetreten. |
| 8253 | 0x0000203D | ERROR_DS_DECODING_ERROR | Ein Decodierungsfehler ist aufgetreten. |
| 8254 | 0x0000203E | ERROR_DS_FILTER_UNKNOWN | Im Such-Filter wird nicht erkannt. |
| 8255 | 0x0000203F | ERROR_DS_PARAM_ERROR | Ein oder mehrere Parameter sind ungültig. |
| 8256 | 0x00002040 | ERROR_DS_NOT_SUPPORTED | Die angegebene Methode wird nicht unterstützt. |
| 8257 | 0x00002041 | ERROR_DS_NO_RESULTS_RETURNED | Keine Ergebnisse wurden zurückgegeben. |
| 8258 | 0x00002042 | ERROR_DS_CONTROL_NOT_FOUND | Das angegebene Steuerelement wird vom Server nicht unterstützt. |
| 8259 | 0x00002043 | ERROR_DS_CLIENT_LOOP | Eine Weiterleitungsschleife wurde vom Client erkannt. |
| 8260 | 0x00002044 | ERROR_DS_REFERRAL_LIMIT_EXCEEDED | Die voreingestellten Empfehlung wurde überschritten. |
| 8261 | 0x00002045 | ERROR_DS_SORT_CONTROL_MISSING | Die Suche erfordert eine SORT-Steuerung. |
| 8262 | 0x00002046 | ERROR_DS_OFFSET_RANGE_ERROR | Die Suchergebnisse überschreiten den angegebenen Offsetbereich. |
| 8301 | 0x0000206D | ERROR_DS_ROOT_MUST_BE_NC | Das Stammobjekt muss der Kopf eines Namenskontextes sein. Das Stammobjekt kann kein instanziiertes übergeordnetes Element haben. |
| 8302 | 0x0000206E | ERROR_DS_ADD_REPLICA_INHIBITED | Add-Replikat-Operation kann nicht ausgeführt werden. Der Namenskontext muss um das Replikat erstellen beschreibbar sein. |
| 8303 | 0x0000206F | ERROR_DS_ATT_NOT_DEF_IN_SCHEMA | Ein Verweis auf ein Attribut, das nicht in das Schema definiert ist aufgetreten. |
| 8304 | 0x00002070 | ERROR_DS_MAX_OBJ_SIZE_EXCEEDED | Die maximale Größe eines Objekts wurde überschritten. |
| 8305 | 0x00002071 | ERROR_DS_OBJ_STRING_NAME_EXISTS | Es wurde versucht, ein Objekt in das Verzeichnis mit einem Namen hinzuzufügen, die bereits verwendet wird. |
| 8306 | 0x00002072 | ERROR_DS_NO_RDN_DEFINED_IN_SCHEMA | Wurde versucht, ein Objekt einer Klasse hinzuzufügen, die keine RDN im Schema definiert. |
| 8307 | 0x00002073 | ERROR_DS_RDN_DOESNT_MATCH_SCHEMA | Wurde versucht, fügen Sie ein Objekt mithilfe eines RDN, die nicht der RDN im Schema definiert ist. |
| 8308 | 0x00002074 | ERROR_DS_NO_REQUESTED_ATTRS_FOUND | Keines der angeforderten Attribute wurden auf die Objekte gefunden. |
| 8309 | 0x00002075 | ERROR_DS_USER_BUFFER_TOO_SMALL | Der Benutzerpuffer ist zu klein. |
| 8310 | 0x00002076 | ERROR_DS_ATT_IS_NOT_ON_OBJ | Das in den Vorgang angegebene Attribut ist nicht vorhanden, auf das Objekt. |
| 8311 | 0x00002077 | ERROR_DS_ILLEGAL_MOD_OPERATION | Illegale Änderungsvorgang. Einige Aspekte der Änderung ist nicht zulässig. |
| 8312 | 0x00002078 | ERROR_DS_OBJ_TOO_LARGE | Das angegebene Objekt ist zu groß. |
| 8313 | 0x00002079 | ERROR_DS_BAD_INSTANCE_TYPE | Der angegebene Instanzentyp ist nicht gültig. |
| 8314 | 0x0000207A | ERROR_DS_MASTERDSA_REQUIRED | Der Vorgang muss an einer master-DSA ausgeführt werden. |
| 8315 | 0x0000207B | ERROR_DS_OBJECT_CLASS_REQUIRED | Das Objekt-Class-Attribut muss angegeben werden. |

| Fehler | | | Beschreibung |
|---------|-------------|-----------------------------------|---|
| Dezimal | Hexadezimal | Name | |
| 8316 | 0x0000207C | ERROR_DS_MISSING_REQUIRED_ATT | Ein erforderliches Attribut fehlt. |
| 8317 | 0x0000207D | ERROR_DS_ATT_NOT_DEF_FOR_CLASS | Es wurde versucht, ein Objekt, um ein Attribut zu enthalten, die nicht legal für seine Klasse ist zu ändern |
| 8318 | 0x0000207E | ERROR_DS_ATT_ALREADY_EXISTS | Das angegebene Attribut ist bereits für das Objekt. |
| 8320 | 0x00002080 | ERROR_DS_CANT_ADD_ATT_VALUES | Das angegebene Attribut ist nicht vorhanden oder hat keine Werte. |
| 8321 | 0x00002081 | ERROR_DS_SINGLE_VALUE_CONSTRAINT | Für ein Attribut, die nur einen Wert haben kann, wurden mehrere Werte angegeben. |
| 8322 | 0x00002082 | ERROR_DS_RANGE_CONSTRAINT | Ein Wert für das Attribut wurde nicht in den zulässigen Wertebereich. |
| 8323 | 0x00002083 | ERROR_DS_ATT_VAL_ALREADY_EXISTS | Der angegebene Wert ist bereits vorhanden. |
| 8324 | 0x00002084 | ERROR_DS_CANT_REM_MISSING_ATT | Das Attribut kann nicht entfernt werden, da sie nicht auf das Objekt vorhanden ist. |
| 8325 | 0x00002085 | ERROR_DS_CANT_REM_MISSING_ATT_VAL | Der Attributwert kann nicht entfernt werden, da sie nicht auf das Objekt vorhanden ist. |
| 8326 | 0x00002086 | ERROR_DS_ROOT_CANT_BE_SUBREF | Das angegebene Stammobjekt kann nicht Objekt einer Subref sein. |
| 8327 | 0x00002087 | ERROR_DS_NO_CHAINING | Verkettung ist nicht gestattet. |
| 8328 | 0x00002088 | ERROR_DS_NO_CHAINED_EVAL | Verkettete Bewertung ist nicht gestattet. |
| 8329 | 0x00002089 | ERROR_DS_NO_PARENT_OBJECT | Der Vorgang konnte nicht ausgeführt werden, weil das Objekt übergeordneten instanziiert oder gelöscht wird. |
| 8330 | 0x0000208A | ERROR_DS_PARENT_IS_AN_ALIAS | Dass ein Elternteil, das ein Alias ist, ist nicht gestattet. Aliase sind Endobjekte. |
| 8331 | 0x0000208B | ERROR_DS_CANT_MIX_MASTER_AND_REPS | Das Objekt und das übergeordnete müssen denselben Typ haben, entweder beide Master oder beide Replikate sein. |
| 8332 | 0x0000208C | ERROR_DS_CHILDREN_EXIST | Der Vorgang kann nicht ausgeführt werden, da untergeordnete Objekte vorhanden sind. Dieser Vorgang kann nur auf einem Blatt-Objekt ausgeführt werden. |
| 8333 | 0x0000208D | ERROR_DS_OBJ_NOT_FOUND | Verzeichnis-Objekt nicht gefunden. |
| 8334 | 0x0000208E | ERROR_DS_ALIASED_OBJ_MISSING | Das Alias-Objekt fehlt. |
| 8335 | 0x0000208F | ERROR_DS_BAD_NAME_SYNTAX | Die zu verwendenden Objektamen ist Syntax nicht. |
| 8336 | 0x00002090 | ERROR_DS_ALIAS_POINTS_TO_ALIAS | Es ist nicht gestattet, für einen Alias auf einen anderen Alias verweisen. |
| 8337 | 0x00002091 | ERROR_DS_CANT_DEREF_ALIAS | Der Alias kann nicht dereferenziert werden. |
| 8338 | 0x00002092 | ERROR_DS_OUT_OF_SCOPE | Die Bedienung ist außerhalb des gültigen Bereichs. |
| 8339 | 0x00002093 | ERROR_DS_OBJECT_BEING_REMOVED | Der Vorgang kann nicht fortgesetzt werden, da das Objekt wird gerade entfernt wird. |
| 8340 | 0x00002094 | ERROR_DS_CANT_DELETE_DSA_OBJ | Das DSA-Objekt kann nicht gelöscht werden. |
| 8341 | 0x00002095 | ERROR_DS_GENERIC_ERROR | Ein Verzeichnisdienstfehler ist aufgetreten. |
| 8342 | 0x00002096 | ERROR_DS_DSA_MUST_BE_INT_MASTER | Der Vorgang kann nur auf einem internen master-DSA-Objekt ausgeführt werden. |
| 8343 | 0x00002097 | ERROR_DS_CLASS_NOT_DSA | Das Objekt muss der Klasse DSA sein. |
| 8344 | 0x00002098 | ERROR_DS_INSUFF_ACCESS_RIGHTS | Unzureichende Berechtigungen zum Ausführen des Vorgangs. |
| 8345 | 0x00002099 | ERROR_DS_ILLEGAL_SUPERIOR | Das Objekt kann nicht hinzugefügt werden, da das übergeordnete Element nicht auf der Liste der möglichen Vorgesetzten ist. |
| 8346 | 0x0000209A | ERROR_DS_ATTRIBUTE_OWNED_BY_SAM | Zugriff auf das Attribut ist unzulässig, da das Attribut von Security Accounts Manager (SAM) gehört. |
| 8347 | 0x0000209B | ERROR_DS_NAME_TOO_MANY_PARTS | Der Name hat zu viele Teile. |
| 8348 | 0x0000209C | ERROR_DS_NAME_TOO_LONG | Der Name ist zu lang. |
| 8349 | 0x0000209D | ERROR_DS_NAME_VALUE_TOO_LONG | Der Namenswert ist zu lang. |
| 8350 | 0x0000209E | ERROR_DS_NAME_UNPARSEABLE | Der Verzeichnisdienst-Fehler analysieren einen Namen. |
| 8351 | 0x0000209F | ERROR_DS_NAME_TYPE_UNKNOWN | Der Verzeichnisdienst kann nicht den Attributtyp für einen Namen erhalten. |

| Fehler | | | Beschreibung |
|---------|-------------|-------------------------------------|---|
| Dezimal | Hexadezimal | Name | |
| 8352 | 0x000020A0 | ERROR_DS_NOT_AN_OBJECT | Der Name identifiziert kein Objekt; der Name bezeichnet ein Phantom. |
| 8353 | 0x000020A1 | ERROR_DS_SEC_DESC_TOO_SHORT | Die Sicherheitsbeschreibung ist zu kurz. |
| 8354 | 0x000020A2 | ERROR_DS_SEC_DESC_INVALID | Die Sicherheitsbeschreibung ist ungültig. |
| 8355 | 0x000020A3 | ERROR_DS_NO_DELETED_NAME | Fehler beim Namen für gelöschte Objekt zu erstellen. |
| 8356 | 0x000020A4 | ERROR_DS_SUBREF_MUST_HAVE_PARENT | Das übergeordnete Objekt einer neuen Subref muss vorhanden sein. |
| 8357 | 0x000020A5 | ERROR_DS_NCNAME_MUST_BE_NC | Das Objekt muss ein Namenskontext sein. |
| 8358 | 0x000020A6 | ERROR_DS_CANT_ADD_SYSTEM_ONLY | Es ist nicht erlaubt, ein Attribut hinzuzufügen, die das System gehört. |
| 8359 | 0x000020A7 | ERROR_DS_CLASS_MUST_BE_CONCRETE | Die Klasse des Objekts muss strukturelle sein; eine abstrakte Klasse kann nicht instanziiert werden. |
| 8360 | 0x000020A8 | ERROR_DS_INVALID_DMD | Das Schema-Objekt konnte nicht gefunden werden. |
| 8361 | 0x000020A9 | ERROR_DS_OBJ_GUID_EXISTS | Ein lokales Objekt mit dieser GUID (tot oder lebendig) ist bereits vorhanden. |
| 8362 | 0x000020AA | ERROR_DS_NOT_ON_BACKLINK | Der Vorgang kann nicht auf einen Backlink ausgeführt werden. |
| 8363 | 0x000020AB | ERROR_DS_NO_CROSSREF_FOR_NC | Der Querverweis für den angegebenen Namenskontext konnte nicht gefunden werden. |
| 8364 | 0x000020AC | ERROR_DS_SHUTTING_DOWN | Der Vorgang konnte nicht ausgeführt werden, da der Verzeichnisdienst heruntergefahren wird. |
| 8365 | 0x000020AD | ERROR_DS_UNKNOWN_OPERATION | Die Verzeichnis-Service-Anforderung ist ungültig. |
| 8366 | 0x000020AE | ERROR_DS_INVALID_ROLE_OWNER | Das Attribut Role Besitzer konnte nicht gelesen werden. |
| 8367 | 0x000020AF | ERROR_DS_COULDNT_CONTACT_FSMO | Der angeforderte FSMO-Vorgang ist fehlgeschlagen. Der aktuelle FSMO-Halter konnten nicht erreicht werden. |
| 8368 | 0x000020B0 | ERROR_DS_CROSS_NC_DN_RENAME | Änderung der DN über einen Namenskontext ist nicht gestattet. |
| 8369 | 0x000020B1 | ERROR_DS_CANT_MOD_SYSTEM_ONLY | Das Attribut kann nicht geändert werden, da es dem System gehört. |
| 8370 | 0x000020B2 | ERROR_DS_REPLICATOR_ONLY | Nur der Replikator kann diese Funktion auszuführen. |
| 8371 | 0x000020B3 | ERROR_DS_OBJ_CLASS_NOT_DEFINED | Die angegebene Klasse ist nicht definiert. |
| 8372 | 0x000020B4 | ERROR_DS_OBJ_CLASS_NOT_SUBCLASS | Die angegebene Klasse ist keine Unterklasse. |
| 8373 | 0x000020B5 | ERROR_DS_NAME_REFERENCE_INVALID | Der Namensbezug ist ungültig. |
| 8374 | 0x000020B6 | ERROR_DS_CROSS_REF_EXISTS | Ein Querverweis ist bereits vorhanden. |
| 8375 | 0x000020B7 | ERROR_DS_CANT_DEL_MASTER_CROSSREF | Es ist nicht gestattet, ein master Querverweis zu löschen. |
| 8376 | 0x000020B8 | ERROR_DS_SUBTREE_NOTIFY_NOT_NC_HEAD | Unterstruktur Benachrichtigungen werden nur von NC-Köpfen unterstützt. |
| 8377 | 0x000020B9 | ERROR_DS_NOTIFY_FILTER_TOO_COMPLEX | Benachrichtigungsfilter ist zu komplex. |
| 8378 | 0x000020BA | ERROR_DS_DUP_RDN | Schemaaktualisierung ist fehlgeschlagen: RDN zu duplizieren. |
| 8379 | 0x000020BB | ERROR_DS_DUP_OID | Schemaaktualisierung ist fehlgeschlagen: OID duplizieren |
| 8380 | 0x000020BC | ERROR_DS_DUP_MAPI_ID | Schemaaktualisierung ist fehlgeschlagen: MAPI-ID doppelt. |
| 8381 | 0x000020BD | ERROR_DS_DUP_SCHEMA_ID_GUID | Schemaaktualisierung ist fehlgeschlagen: doppelte Schema-Id-GUID. |
| 8382 | 0x000020BE | ERROR_DS_DUP_LDAP_DISPLAY_NAME | Schemaaktualisierung ist fehlgeschlagen: doppelte LDAP-Anzeigename. |
| 8383 | 0x000020BF | ERROR_DS_SEMANTIC_ATT_TEST | Schemaaktualisierung ist fehlgeschlagen: Spektrum-untere weniger als obere Bereich |
| 8384 | 0x000020C0 | ERROR_DS_SYNTAX_MISMATCH | Schemaaktualisierung ist fehlgeschlagen: Syntax-Konflikt |
| 8385 | 0x000020C1 | ERROR_DS_EXISTS_IN_MUST_HAVE | Löschen des Schemas ist fehlgeschlagen: Attribut wird verwendet in muss enthalten |
| 8386 | 0x000020C2 | ERROR_DS_EXISTS_IN_MAY_HAVE | Löschen des Schemas ist fehlgeschlagen: Attribut wird verwendet, im Mai-enhalten |

| Fehler | | | Beschreibung |
|---------|-------------|-----------------------------------|--|
| Dezimal | Hexadezimal | Name | |
| 8387 | 0x000020C3 | ERROR_DS_NONEXISTENT_MAY_HAVE | Schemaaktualisierung ist fehlgeschlagen: Attribut in Mai-enthalten ist nicht vorhanden |
| 8388 | 0x000020C4 | ERROR_DS_NONEXISTENT_MUST_HAVE | Schemaaktualisierung ist fehlgeschlagen: Attribut in muss enthalten-existiert nicht |
| 8389 | 0x000020C5 | ERROR_DS_AUX_CLS_TEST_FAIL | Schemaaktualisierung ist fehlgeschlagen: Klasse in der Aux-Klasse Liste existiert nicht oder ist keine Erweiterungsklasse |
| 8390 | 0x000020C6 | ERROR_DS_NONEXISTENT_POSS_SUP | Schemaaktualisierung ist fehlgeschlagen: Klasse in Poss-Vorgesetzten ist nicht vorhanden |
| 8391 | 0x000020C7 | ERROR_DS_SUB_CLS_TEST_FAIL | Schemaaktualisierung ist fehlgeschlagen: Klasse in der Subclassof-Liste ist nicht vorhanden oder erfüllt nicht Hierarchieregeln |
| 8392 | 0x000020C8 | ERROR_DS_BAD_RDN_ATT_ID_SYNTAX | Schemaaktualisierung ist fehlgeschlagen: Rdn-Att-Id hat falsche Syntax |
| 8393 | 0x000020C9 | ERROR_DS_EXISTS_IN_AUX_CLS | Löschen des Schemas ist fehlgeschlagen: Klasse wird als Erweiterungsklasse verwendet |
| 8394 | 0x000020CA | ERROR_DS_EXISTS_IN_SUB_CLS | Löschen des Schemas ist fehlgeschlagen: Klasse wird als Sub-Klasse verwendet |
| 8395 | 0x000020CB | ERROR_DS_EXISTS_IN_POSS_SUP | Löschen des Schemas ist fehlgeschlagen: Klasse wird verwendet, als Poss-Superior |
| 8396 | 0x000020CC | ERROR_DS_RECALCSHEMA_FAILED | Schemaaktualisierung ist fehlgeschlagen in Validierung Cache eine Neuberechnung. |
| 8397 | 0x000020CD | ERROR_DS_TREE_DELETE_NOT_FINISHED | Der Strukturlöschung ist noch nicht fertig. |
| 8398 | 0x000020CE | ERROR_DS_CANT_DELETE | Der angeforderte Löschvorgang konnte nicht ausgeführt werden. |
| 8399 | 0x000020CF | ERROR_DS_ATT_SCHEMA_REQ_ID | Kann nicht gelesen werden die Klassen-ID für das Schema-Protokoll regelt. |
| 8400 | 0x000020D0 | ERROR_DS_BAD_ATT_SCHEMA_SYNTAX | Das Attributschema ist Syntax nicht. |
| 8401 | 0x000020D1 | ERROR_DS_CANT_CACHE_ATT | Das Attribut konnte nicht zwischengespeichert werden. |
| 8402 | 0x000020D2 | ERROR_DS_CANT_CACHE_CLASS | Die Klasse konnte nicht zwischengespeichert werden. |
| 8403 | 0x000020D3 | ERROR_DS_CANT_REMOVE_ATT_CACHE | Das Attribut konnte nicht aus dem Cache entfernt werden. |
| 8404 | 0x000020D4 | ERROR_DS_CANT_REMOVE_CLASS_CACHE | Die Klasse konnte nicht aus dem Cache entfernt werden. |
| 8405 | 0x000020D5 | ERROR_DS_CANT_RETRIEVE_DN | Das distinguished Name-Attribut konnte nicht gelesen werden. |
| 8406 | 0x000020D6 | ERROR_DS_MISSING_SUPREF | Objekt eine erforderliche Subref fehlt. |
| 8407 | 0x000020D7 | ERROR_DS_CANT_RETRIEVE_INSTANCE | Das Type-Instanzattribut konnte nicht abgerufen werden. |
| 8408 | 0x000020D8 | ERROR_DS_CODE_INCONSISTENCY | Ein interner Fehler aufgetreten. |
| 8409 | 0x000020D9 | ERROR_DS_DATABASE_ERROR | Ein Datenbankfehler ist aufgetreten. |
| 8410 | 0x000020DA | ERROR_DS_GOVERNSID_MISSING | Das Attribut GOVERNSID fehlt. |
| 8411 | 0x000020DB | ERROR_DS_MISSING_EXPECTED_ATT | Eine erwartete Attribut fehlt. |
| 8412 | 0x000020DC | ERROR_DS_NCNAME_MISSING_CR_REF | Im angegebene Namenskontext fehlt ein Querverweis. |
| 8413 | 0x000020DD | ERROR_DS_SECURITY_CHECKING_ERROR | Eine Sicherheitsüberprüfung-Fehler ist aufgetreten. |
| 8414 | 0x000020DE | ERROR_DS_SCHEMA_NOT_LOADED | Das Schema ist nicht geladen. |
| 8415 | 0x000020DF | ERROR_DS_SCHEMA_ALLOC_FAILED | Schema Zuweisung ist fehlgeschlagen. Bitte prüfen Sie, ob die Maschine Arbeitsspeicher knapp ist. |
| 8416 | 0x000020E0 | ERROR_DS_ATT_SCHEMA_REQ_SYNTAX | Erhalten Sie die erforderliche Syntax für das Attributschema konnte nicht. |
| 8417 | 0x000020E1 | ERROR_DS_GCVERIFY_ERROR | Die Überprüfung des globalen Katalogs ist fehlgeschlagen. Der globale Katalog ist nicht verfügbar oder unterstützt nicht den Betrieb. Ein Teil des Verzeichnisses ist zurzeit nicht verfügbar. |
| 8418 | 0x000020E2 | ERROR_DS_DRA_SCHEMA_MISMATCH | Der Replikationsvorgang ist aufgrund einer Schemaübereinstimmung zwischen den beteiligten Servern fehlgeschlagen. |
| 8419 | 0x000020E3 | ERROR_DS_CANT_FIND_DSA_OBJ | Das DSA-Objekt konnte nicht gefunden werden. |
| 8420 | 0x000020E4 | ERROR_DS_CANT_FIND_EXPECTED_NC | Der Namenskontext konnte nicht gefunden werden. |

| Fehler | | | Beschreibung |
|---------|-------------|--|---|
| Dezimal | Hexadezimal | Name | |
| 8421 | 0x000020E5 | ERROR_DS_CANT_FIND_NC_IN_CACHE | Der Namenskontext konnte nicht im Cache gefunden werden. |
| 8422 | 0x000020E6 | ERROR_DS_CANT_RETRIEVE_CHILD | Das untergeordnete Objekt konnte nicht abgerufen werden. |
| 8423 | 0x000020E7 | ERROR_DS_SECURITY_ILLEGAL_MODIFY | Die Änderung war aus Sicherheitsgründen nicht gestattet. |
| 8424 | 0x000020E8 | ERROR_DS_CANT_REPLACE_HIDDEN_REC | Der Vorgang kann nicht den versteckten Eintrag ersetzen. |
| 8425 | 0x000020E9 | ERROR_DS_BAD_HIERARCHY_FILE | Die Hierarchie-Datei ist ungültig. |
| 8426 | 0x000020EA | ERROR_DS_BUILD_HIERARCHY_TABLE_FAILED | Der Versuch, die Hierarchietabelle zu erstellen gescheitert. |
| 8427 | 0x000020EB | ERROR_DS_CONFIG_PARAM_MISSING | Der Verzeichnis-Konfigurations-Parameter fehlt in der Registrierung. |
| 8428 | 0x000020EC | ERROR_DS_COUNTING_AB_INDICES_FAILED | Der Versuch, die Adresse Buch Indizes zählen konnte. |
| 8429 | 0x000020ED | ERROR_DS_HIERARCHY_TABLE_MALLOC_FAILED | Die Reservierung der Hierarchietabelle ist fehlgeschlagen. |
| 8430 | 0x000020EE | ERROR_DS_INTERNAL_FAILURE | Der Verzeichnisdienst ist einen internen Fehler aufgetreten. |
| 8431 | 0x000020EF | ERROR_DS_UNKNOWN_ERROR | Der Verzeichnisdienst ist einen unbekanntem Fehler aufgetreten. |
| 8432 | 0x000020F0 | ERROR_DS_ROOT_REQUIRES_CLASS_TOP | Ein Stammobjekt erfordert eine Klasse der 'Top'. |
| 8433 | 0x000020F1 | ERROR_DS_REFUSING_FSMO_ROLES | Dieser Verzeichnisserver wird heruntergefahren und kann nicht übernehmen des Besitzes von neuen schwimmenden Einzelmaster-Operation-Rollen. |
| 8434 | 0x000020F2 | ERROR_DS_MISSING_FSMO_SETTINGS | Der Verzeichnisdienst fehlen verbindliche Konfigurationsinformationen, und ist nicht in der Lage, die Besitzer von schwimmenden Einzelmaster-Operation-Rollen zu bestimmen. |
| 8435 | 0x000020F3 | ERROR_DS_UNABLE_TO_SURRENDER_ROLES | Der Verzeichnisdienst konnte den Besitz von einem oder mehreren schwimmenden Einzelmaster-Operation Rollen an andere Server übertragen. |
| 8436 | 0x000020F4 | ERROR_DS_DRA_GENERIC | Der Replikationsvorgang ist fehlgeschlagen. |
| 8437 | 0x000020F5 | ERROR_DS_DRA_INVALID_PARAMETER | Ein ungültiger Parameter wurde für diesen Replikationsvorgang angegeben. |
| 8438 | 0x000020F6 | ERROR_DS_DRA_BUSY | Der Verzeichnisdienst ist zu beschäftigt, um den Replikationsvorgang zu diesem Zeitpunkt abgeschlossen. |
| 8439 | 0x000020F7 | ERROR_DS_DRA_BAD_DN | Der distinguished Name für diesen Replikationsvorgang angegebene ist ungültig. |
| 8440 | 0x000020F8 | ERROR_DS_DRA_BAD_NC | Der für diesen Replikationsvorgang angegebene Namenskontext ist ungültig. |
| 8441 | 0x000020F9 | ERROR_DS_DRA_DN_EXISTS | Der distinguished Name für diesen Replikationsvorgang bereits angegeben vorhanden ist. |
| 8442 | 0x000020FA | ERROR_DS_DRA_INTERNAL_ERROR | Im Replikationssystem ist einen internen Fehler aufgetreten. |
| 8443 | 0x000020FB | ERROR_DS_DRA_INCONSISTENT_DIT | Der Replikationsvorgang ist eine Datenbankinkonsistenz aufgetreten. |
| 8444 | 0x000020FC | ERROR_DS_DRA_CONNECTION_FAILED | Der für diesen Replikationsvorgang angegebene Server konnte nicht kontaktiert werden. |
| 8445 | 0x000020FD | ERROR_DS_DRA_BAD_INSTANCE_TYPE | Der Replikationsvorgang hat ein Objekt mit einem ungültigen Instanzentyp feststellt. |
| 8446 | 0x000020FE | ERROR_DS_DRA_OUT_OF_MEM | Der Replikationsvorgang ist fehlgeschlagen, Speicher. |
| 8447 | 0x000020FF | ERROR_DS_DRA_MAIL_PROBLEM | Der Replikationsvorgang wurde ein Fehler in das Mailsystem. |
| 8448 | 0x00002100 | ERROR_DS_DRA_REF_ALREADY_EXISTS | Die Replikationsverweisinformationen für den Zielsever bereits vorhanden ist. |
| 8449 | 0x00002101 | ERROR_DS_DRA_REF_NOT_FOUND | Die Replikationsverweisinformationen für den Zielsever ist nicht vorhanden. |
| 8450 | 0x00002102 | ERROR_DS_DRA_OBJ_IS_REP_SOURCE | Der Namenskontext kann nicht entfernt werden, da es auf einen anderen Server repliziert werden. |

| Fehler | | | Beschreibung |
|---------|-------------|--|---|
| Dezimal | Hexadezimal | Name | |
| 8451 | 0x00002103 | ERROR_DS_DRA_DB_ERROR | Der Replikationsvorgang ist einen Datenbankfehler aufgetreten. |
| 8452 | 0x00002104 | ERROR_DS_DRA_NO_REPLICA | Der Namenskontext wird gerade entfernt oder wird nicht vom angegebenen Server repliziert. |
| 8453 | 0x00002105 | ERROR_DS_DRA_ACCESS_DENIED | Der Replikationszugriff wurde verweigert. |
| 8454 | 0x00002106 | ERROR_DS_DRA_NOT_SUPPORTED | Der angeforderte Vorgang wird von dieser Version des Verzeichnisdienstes nicht unterstützt. |
| 8455 | 0x00002107 | ERROR_DS_DRA_RPC_CANCELLED | Der Remoteprozeduraufruf Replikation wurde abgebrochen. |
| 8456 | 0x00002108 | ERROR_DS_DRA_SOURCE_DISABLED | Der Quellserver weist derzeit Replikationsanforderungen. |
| 8457 | 0x00002109 | ERROR_DS_DRA_SINK_DISABLED | Der Zielserver ist derzeit Replikationsanforderungen ablehnen. |
| 8458 | 0x0000210A | ERROR_DS_DRA_NAME_COLLISION | Der Replikationsvorgang ist aufgrund einer Kollision von Objektnamen fehlgeschlagen. |
| 8459 | 0x0000210B | ERROR_DS_DRA_SOURCE_REINSTALLED | Die Replikationsquelle wurde erneut installiert wurde. |
| 8460 | 0x0000210C | ERROR_DS_DRA_MISSING_PARENT | Der Replikationsvorgang ist fehlgeschlagen, da ein erforderliches übergeordnetes Objekt fehlt. |
| 8461 | 0x0000210D | ERROR_DS_DRA_PREEMPTED | Der Replikationsvorgang wurde präemptiv verdrängt. |
| 8462 | 0x0000210E | ERROR_DS_DRA_ABANDON_SYNC | Der Replikationssynchronisierungsversuch wurde aus Mangel an Updates aufgegeben. |
| 8463 | 0x0000210F | ERROR_DS_DRA_SHUTDOWN | Der Replikationsvorgang wurde abgebrochen, da das System heruntergefahren wird. |
| 8464 | 0x00002110 | ERROR_DS_DRA_INCOMPATIBLE_PARTIAL_SE
T | Der Replikationssynchronisierungsversuch ist fehlgeschlagen, da der Ziel-Teilattributsatz keine Teilmenge der Quelle Teilattributsatz. |
| 8465 | 0x00002111 | ERROR_DS_DRA_SOURCE_IS_PARTIAL_REPLI
CA | Der Replikationssynchronisierungsversuch ist fehlgeschlagen, da ein Masterreplikat versucht, aus einem partiellen Replikat synchronisiert. |
| 8466 | 0x00002112 | ERROR_DS_DRA_EXTN_CONNECTION_FAILED | Der für diesen Replikationsvorgang angegebene Server wurde kontaktiert, aber diesen Server konnte keine Verbindung einen zusätzlichen Server benötigt, um den Vorgang abzuschließen. |
| 8467 | 0x00002113 | ERROR_DS_INSTALL_SCHEMA_MISMATCH | Die Version des Active Directory-Schemas der Quellgesamtstruktur ist nicht kompatibel mit der Version von Active Directory auf diesem Computer. Sie müssen das Betriebssystem auf einem Domänencontroller in der Quellgesamtstruktur aktualisieren, bevor dieser Computer als Domänencontroller in dieser Gesamtstruktur hinzugefügt werden kann. |
| 8468 | 0x00002114 | ERROR_DS_DUP_LINK_ID | Schemaaktualisierung ist fehlgeschlagen: Attribut mit der gleichen Verbindungskennung bereits vorhanden ist. |
| 8469 | 0x00002115 | ERROR_DS_NAME_ERROR_RESOLVING | Namensübersetzung: allgemeiner Verarbeitungsfehler. |
| 8470 | 0x00002116 | ERROR_DS_NAME_ERROR_NOT_FOUND | Namensübersetzung: konnte nicht gefunden werden, den Namen oder die unzureichende Recht, Namen zu sehen. |
| 8471 | 0x00002117 | ERROR_DS_NAME_ERROR_NOT_UNIQUE | Namensübersetzung: Input Name mehreren Ausgabenamen zugeordnet. |
| 8472 | 0x00002118 | ERROR_DS_NAME_ERROR_NO_MAPPING | Namensübersetzung: Input Name gefunden, aber nicht die zugehörigen Ausgabe-Format. |
| 8473 | 0x00002119 | ERROR_DS_NAME_ERROR_DOMAIN_ONLY | Namensübersetzung: nicht vollständig lösen nur die Domain wurde gefunden. |
| 8474 | 0x0000211A | ERROR_DS_NAME_ERROR_NO_SYNTACTICAL_
MAPPING | Namensübersetzung: nicht in der Lage, rein syntaktische Zuordnung auf dem Client auszuführen, ohne auf den Draht ausgehen. |
| 8475 | 0x0000211B | ERROR_DS_CONSTRUCTED_ATT_MOD | Änderung der konstruierten Att ist nicht zulässig. |
| 8476 | 0x0000211C | ERROR_DS_WRONG_OM_OBJ_CLASS | Die OM-Objektklasse angegeben ist für ein Attribut mit der angegebenen Syntax falsch. |
| 8477 | 0x0000211D | ERROR_DS_DRA_REPL_PENDING | Die Replikationsanforderung wurde gebucht; warten auf Antwort. |
| 8478 | 0x0000211E | ERROR_DS_DS_REQUIRED | Der angeforderte Vorgang erfordert einen Verzeichnisdienst, und keiner war verfügbar. |

| Fehler | | | Beschreibung |
|---------|-------------|--|---|
| Dezimal | Hexadezimal | Name | |
| 8479 | 0x0000211F | ERROR_DS_INVALID_LDAP_DISPLAY_NAME | Der angezeigte LDAP-Name der Klasse oder des Attributs enthält ASCII fremde Zeichen. |
| 8480 | 0x00002120 | ERROR_DS_NON_BASE_SEARCH | Der angeforderte Suchvorgang wird nur für base-Suche unterstützt. |
| 8481 | 0x00002121 | ERROR_DS_CANT_RETRIEVE_ATTS | Die Suche konnte nicht Attribute aus der Datenbank abzurufen. |
| 8482 | 0x00002122 | ERROR_DS_BACKLINK_WITHOUT_LINK | Der Schema Update-Vorgang versucht, ein Backwardlink-Attribut hinzuzufügen, die keine entsprechenden Forwardlink aufweist. |
| 8483 | 0x00002123 | ERROR_DS_EPOCH_MISMATCH | Quelle und Ziel einer domänenübergreifenden Verschiebung einverstanden nicht auf die Objektnummer Epoche. Quelle oder das Ziel muss nicht die neueste Version des Objekts. |
| 8484 | 0x00002124 | ERROR_DS_SRC_NAME_MISMATCH | Quelle und Ziel einer domänenübergreifenden Verschiebung einverstanden nicht auf der aktuellen Objektname. Quelle oder das Ziel muss nicht die neueste Version des Objekts. |
| 8485 | 0x00002125 | ERROR_DS_SRC_AND_DST_NC_IDENTICAL | Quelle und Ziel einer domänenübergreifenden Verschiebevorgangs sind identisch. Anrufer sollten lokale Verschiebevorgang anstelle von cross Domain Verschiebevorgang verwenden. |
| 8486 | 0x00002126 | ERROR_DS_DST_NC_MISMATCH | Quelle und Ziel einer domänenübergreifenden Verschiebung sind auf die Namenskontexte in der Gesamtstruktur nicht einig. Quelle oder das Ziel muss nicht die neueste Version von den Partitionscontainer. |
| 8487 | 0x00002127 | ERROR_DS_NOT_AUTHORITY_FOR_DST_NC | Ziel einer domänenübergreifenden Verschiebung ist nicht autorisierend für den Ziel-Namenskontext. |
| 8488 | 0x00002128 | ERROR_DS_SRC_GUID_MISMATCH | Quelle und Ziel einer domänenübergreifenden Verschiebung einverstanden nicht über die Identität des Quellobjekts. Quelle oder das Ziel muss nicht die neueste Version des Quellobjekts. |
| 8489 | 0x00002129 | ERROR_DS_CANT_MOVE_DELETED_OBJECT | Objekt, das domänenübergreifend verschoben werden ist bereits bekannt, vom Zielserver gelöscht werden. Der Quellserver muss nicht die neueste Version des Quellobjekts. |
| 8490 | 0x0000212A | ERROR_DS_PDC_OPERATION_IN_PROGRESS | Ein anderer Vorgang, der exklusiven Zugriff auf die PDC PSMO benötigt, ist bereits im Gange. |
| 8491 | 0x0000212B | ERROR_DS_CROSS_DOMAIN_CLEANUP_REQD | Eines domänenübergreifenden Verschiebevorgangs ist fehlgeschlagen, so dass die zwei Versionen des verschobenen Objekts vorhanden - jeweils in den Quell- und Ziel-Domänen. Das Zielobjekt muss entfernt werden, um das System in einen konsistenten Zustand wiederherzustellen. |
| 8492 | 0x0000212C | ERROR_DS_ILLEGAL_XDOM_MOVE_OPERATION | Dieses Objekt kann nicht verschoben werden über Domänengrenzen entweder weil domänenübergreifenden bewegt für diese Klasse sind nicht zulässig, oder das Objekt einige Besonderheiten, z. B. hat: Treuhandkonto oder eingeschränkt RID, die verhindern, dass seine Bewegung. |
| 8493 | 0x0000212D | ERROR_DS_CANT_WITH_ACCT_GROUP_MEMBERSHPS | Objekte mit Mitgliedschaften kann nicht über Domänengrenzen verschoben werden, da einmal bewegt, das Bedingungen für die Mitgliedschaft von der Gruppe der verletzen würde. Entfernen Sie das Objekt von jedem Konto Gruppenmitgliedschaften und wiederholen. |
| 8494 | 0x0000212E | ERROR_DS_NC_MUST_HAVE_NC_PARENT | Ein naming Kontext Kopf muss die unmittelbar untergeordneten ein weiteres naming Kontext Kopf, keinen inneren Knoten sein. |

| Fehler | | | Beschreibung |
|---------|-------------|---|--|
| Dezimal | Hexadezimal | Name | |
| 8495 | 0x0000212F | ERROR_DS_CR_IMPOSSIBLE_TO_VALIDATE | Das Verzeichnis kann nicht den vorgeschlagenen Kontextnamen für naming überprüfen, weil es kein Replikat des Namenskontextes oberhalb des vorgeschlagenen Namenskontextes hält. Bitte stellen Sie sicher, dass die Domänennamenmaster-Rolle von einem Server stattfindet, die als globale Katalogserver konfiguriert ist und der Server auf dem neuesten Stand der Replikationspartner ist. (Gilt nur für Windows 2000 Domain Naming Master) |
| 8496 | 0x00002130 | ERROR_DS_DST_DOMAIN_NOT_NATIVE | Zieldomäne muss sich im einheitlichen Modus. |
| 8497 | 0x00002131 | ERROR_DS_MISSING_INFRASTRUCTURE_CONTAINER | Der Vorgang kann nicht ausgeführt werden, weil der Server nicht verfügbar einen Infrastruktur-Container in der Domäne von Interesse sind. |
| 8498 | 0x00002132 | ERROR_DS_CANT_MOVE_ACCOUNT_GROUP | Domänenübergreifende Verschieben von nicht leeren Kontogruppen ist nicht erlaubt. |
| 8499 | 0x00002133 | ERROR_DS_CANT_MOVE_RESOURCE_GROUP | Domänenübergreifende Verschieben von nicht leeren Ressourcengruppen ist nicht erlaubt. |

| Fehler | | | Beschreibung |
|---------|-------------|---|---|
| Dezimal | Hexadezimal | Name | |
| 8500 | 0x00002134 | ERROR_DS_INVALID_SEARCH_FLAG | Die Search-Flags für das Attribut sind ungültig. Das ANR-Bit ist nur für Attribute von Unicode- oder Teletex-Zeichenfolgen gültig. |
| 8501 | 0x00002135 | ERROR_DS_NO_TREE_DELETE_ABOVE_NC | Baum-Löschungen, die an ein-mit einem NC-Kopf als abhängiges Objekt beginnen, sind nicht zulässig. |
| 8502 | 0x00002136 | ERROR_DS_COULDNT_LOCK_TREE_FOR_DELETE | Der Verzeichnisdienst konnte eine Struktur zur Vorbereitung einer Strukturlöschung zu sperren, da die Struktur verwendet wurde. |
| 8503 | 0x00002137 | ERROR_DS_COULDNT_IDENTIFY_OBJECTS_FOR_TREE_DELETE | Der Verzeichnisdienst konnte die Liste der zu löschenden beim Versuch einer Strukturlöschung Objekte zu identifizieren. |
| 8504 | 0x00002138 | ERROR_DS_SAM_INIT_FAILURE | Sicherheitskontenverwaltung ist aufgrund des folgenden Fehlers fehlgeschlagen: %1.
Fehlerstatus: 0 x %2. Klicken Sie auf OK, um das System Herunterfahren und Neustart in Verzeichnisdienste wiederherstellen. Überprüfen Sie das Ereignisprotokoll detaillierte Informationen. |
| 8505 | 0x00002139 | ERROR_DS_SENSITIVE_GROUP_VIOLATION | Nur ein Administrator kann die Liste der Mitglieder einer administrativen Gruppe ändern. |
| 8506 | 0x0000213A | ERROR_DS_CANT_MOD_PRIMARYGROUPID | Die primäre Gruppenkennung eines Domänencontrollerkontos kann nicht geändert werden. |
| 8507 | 0x0000213B | ERROR_DS_ILLEGAL_BASE_SCHEMA_MOD | Wird versucht, das Basisschema zu ändern. |
| 8508 | 0x0000213C | ERROR_DS_NONSAFE_SCHEMA_CHANGE | Hinzufügen eines neuen obligatorischen Attributs zu einer vorhandenen Klasse, ein obligatorisches Attribut aus einer vorhandenen Klasse löschen oder Hinzufügen von ein optionales Attribut auf Sonderklasse ist Spitze, die kein Backlink Attribut (direkt oder durch Vererbung, z. B. durch Hinzufügen oder Löschen einer Erweiterungsklasse) ist nicht zulässig. |
| 8509 | 0x0000213D | ERROR_DS_SCHEMA_UPDATE_DISALLOWED | Schemaaktualisierung ist auf diesem DC nicht zulässig, weil der DC nicht das Schema-FSMO-Funktionsbesitzer ist. |
| 8510 | 0x0000213E | ERROR_DS_CANT_CREATE_UNDER_SCHEMA | Ein Objekt dieser Klasse kann unter dem Schemacontainer nicht erstellt werden. Sie können nur Objekte unter dem Schemacontainer Attributschema und Klasse-Schema erstellen. |
| 8511 | 0x0000213F | ERROR_DS_INSTALL_NO_SRC_SCH_VERSION | Installation des Replikats bzw. untergeordneten konnte das Attribut ObjectVersion auf dem Schemacontainer auf der Quell-DC. Entweder fehlt auf dem Schemacontainer das Attribut, oder die angegebenen Anmeldeinformationen sind nicht berechtigt, es zu lesen. |
| 8512 | 0x00002140 | ERROR_DS_INSTALL_NO_SCH_VERSION_IN_IFILE | Die Installation des Replikats bzw. untergeordneten konnte das Attribut ObjectVersion im Abschnitt SCHEMA der Datei im system32-Verzeichnis. |
| 8513 | 0x00002141 | ERROR_DS_INVALID_GROUP_TYPE | Der angegebene Gruppentyp ist ungültig. |
| 8514 | 0x00002142 | ERROR_DS_NO_NEST_GLOBALGROUP_IN_MIXEDDOMAIN | Globale Gruppen in einer Domäne im gemischten Modus können nicht verschachtelt werden, wenn die Gruppe Sicherheit aktiviert ist. |
| 8515 | 0x00002143 | ERROR_DS_NO_NEST_LOCALGROUP_IN_MIXEDDOMAIN | Lokale Gruppen in einer Domäne im gemischten Modus können nicht verschachtelt werden, wenn die Gruppe Sicherheit aktiviert ist. |
| 8516 | 0x00002144 | ERROR_DS_GLOBAL_CANT_HAVE_LOCAL_MEMBER | Eine globale Gruppe nicht als Mitglied eine lokale Gruppe. |
| 8517 | 0x00002145 | ERROR_DS_GLOBAL_CANT_HAVE_UNIVERSAL_MEMBER | Eine globale Gruppe kann keine universelle Gruppe als Mitglied haben. |
| 8518 | 0x00002146 | ERROR_DS_UNIVERSAL_CANT_HAVE_LOCAL_MEMBER | Eine universelle Gruppe kann nicht als Mitglied eine lokale Gruppe haben. |
| 8519 | 0x00002147 | ERROR_DS_GLOBAL_CANT_HAVE_CROSSDOMAIN_MEMBER | Eine globale Gruppe kann nicht Mitglied domänenübergreifende haben. |
| 8520 | 0x00002148 | ERROR_DS_LOCAL_CANT_HAVE_CROSSDOMAIN_LOCAL_MEMBER | Eine lokale Gruppe kann nicht als Mitglied einer anderen domänenübergreifende lokale Gruppe haben. |

| Fehler | | | Beschreibung |
|--------------|-------------|--|--|
| Dezi-
mal | Hexadezimal | Name | |
| 8521 | 0x00002149 | ERROR_DS_HAVE_PRIMARY_MEMBERS | Eine Gruppe mit primären Mitgliedern kann nicht in eine Gruppe mit deaktivierter Sicherheit ändern. |
| 8522 | 0x0000214A | ERROR_DS_STRING_SD_CONVERSION_FAILED | Der Cacheload des Schemas konnte die Zeichenfolge Standard-SD bei einem Klassenschemaobjekt konvertieren. |
| 8523 | 0x0000214B | ERROR_DS_NAMING_MASTER_GC | Nur DSAs, die Globalkatalogserver konfiguriert darf die Domain Naming Master FSMO-Rolle zu halten. (Gilt nur für Windows 2000 Server) |
| 8524 | 0x0000214C | ERROR_DS_LOOKUP_FAILURE | Der DSA-Vorgang kann aufgrund eines DNS-Lookup nicht fortfahren. |
| 8525 | 0x0000214D | ERROR_DS_COULDNT_UPDATE_SPNS | Beim Verarbeiten einer Änderung des DNS-Hostnamens für ein Objekt, konnte der Dienstprinzipalnamens-Werte nicht synchron gehalten werden. |
| 8526 | 0x0000214E | ERROR_DS_CANT_RETRIEVE_SD | Das Attribut Sicherheitsbeschreibung konnte nicht gelesen werden. |
| 8527 | 0x0000214F | ERROR_DS_KEY_NOT_UNIQUE. | Das angeforderte Objekt wurde nicht gefunden, aber ein Objekt mit diesem Schlüssel wurde gefunden. |
| 8528 | 0x00002150 | ERROR_DS_WRONG_LINKED_ATT_SYNTAX | Die Syntax für die verknüpfte attributierten hinzugefügt wird, ist falsch. Forwardlinks haben nur die Syntax 2.5.5.1, 2.5.5.7 und 2.5.5.14 und Backlinks nur die Syntax 2.5.5.1 möglich. |
| 8529 | 0x00002151 | ERROR_DS_SAM_NEED_BOOTKEY_PASSWOR
D | Die Sicherheitskontenverwaltung muss das Boot-Passwort zu erhalten. |
| 8530 | 0x00002152 | ERROR_DS_SAM_NEED_BOOTKEY_FLOPPY | Die Sicherheitskontenverwaltung muss den Schlüssel Boot von Diskette zu erhalten. |
| 8531 | 0x00002153 | ERROR_DS_CANT_START | Verzeichnisdienst kann nicht gestartet werden. |
| 8532 | 0x00002154 | ERROR_DS_INIT_FAILURE | Verzeichnisdienste konnten nicht gestartet werden. |
| 8533 | 0x00002155 | ERROR_DS_NO_PKT_PRIVACY_ON_CONNECTI
ON | Die Verbindung zwischen Client und Server erfordert Paketsicherheit oder besser. |
| 8534 | 0x00002156 | ERROR_DS_SOURCE_DOMAIN_IN_FOREST | Die Quelldomäne möglicherweise nicht in der gleichen Gesamtstruktur als Ziel. |
| 8535 | 0x00002157 | ERROR_DS_DESTINATION_DOMAIN_NOT_IN_F
OREST | Die Zieldomäne muss sich im Wald. |
| 8536 | 0x00002158 | ERROR_DS_DESTINATION_AUDITING_NOT_EN
ABLED | Der Vorgang erfordert, dass Ziel Domain Überwachung aktiviert werden. |
| 8537 | 0x00002159 | ERROR_DS_CANT_FIND_DC_FOR_SRC_DOMAI
N | Der Vorgang konnte keinen DC für die Quelldomäne finden. |
| 8538 | 0x0000215A | ERROR_DS_SRC_OBJ_NOT_GROUP_OR_USER | Das Quellobjekt muss eine Gruppe oder einen Benutzer sein. |
| 8539 | 0x0000215B | ERROR_DS_SRC_SID_EXISTS_IN_FOREST | SID des Quellobjekts ist bereits in Ziel-Gesamtstruktur. |
| 8540 | 0x0000215C | ERROR_DS_SRC_AND_DST_OBJECT_CLASS_M
ISMATCH | Das Quell- und Ziel-Objekt muss vom gleichen Typ sein. |
| 8541 | 0x0000215D | ERROR_SAM_INIT_FAILURE | Sicherheitskontenverwaltung ist aufgrund des folgenden Fehlers fehlgeschlagen: %1. Fehlerstatus: 0 x %2. Klicken Sie auf OK, um das System Herunterfahren und starten Sie im abgesicherten Modus neu. Überprüfen Sie das Ereignisprotokoll detaillierte Informationen. |
| 8542 | 0x0000215E | ERROR_DS_DRA_SCHEMA_INFO_SHIP | Schemainformationen konnten nicht in die Replikationsanforderung einbezogen. |
| 8543 | 0x0000215F | ERROR_DS_DRA_SCHEMA_CONFLICT | Der Replikationsvorgang konnte nicht abgeschlossen werden, aufgrund einer Inkompatibilität Schema. |
| 8544 | 0x00002160 | ERROR_DS_DRA_EARLIER_SCHEMA_CONFLICT | Der Replikationsvorgang konnte nicht abgeschlossen werden, aufgrund einer vorherigen Schema-Inkompatibilität. |
| 8545 | 0x00002161 | ERROR_DS_DRA_OBJ_NC_MISMATCH | Die Replikationsaktualisierung konnte nicht angewendet werden, da die Quelle oder das Ziel hat, noch nicht erhaltenen Informationen bezüglich eines letzten domänenübergreifenden verschieben-Operation. |

| Fehler | | | Beschreibung |
|---------|-------------|---|---|
| Dezimal | Hexadezimal | Name | |
| 8546 | 0x00002162 | ERROR_DS_NC_STILL_HAS_DSAS | Die angeforderte Domäne konnte nicht gelöscht werden, denn es gibt ein Domänencontroller, die immer noch diese Domain hosten. |
| 8547 | 0x00002163 | ERROR_DS_GC_REQUIRED | Der angeforderte Vorgang kann nur auf einem globalen Katalogserver ausgeführt werden. |
| 8548 | 0x00002164 | ERROR_DS_LOCAL_MEMBER_OF_LOCAL_ONLY | Eine lokale Gruppe kann nur Mitglied in anderen lokalen Gruppen in derselben Domäne sein. |
| 8549 | 0x00002165 | ERROR_DS_NO_FPO_IN_UNIVERSAL_GROUPS | Fremde Sicherheitsprinzipale können keine Mitglieder von universellen Gruppen sein. |
| 8550 | 0x00002166 | ERROR_DS_CANT_ADD_TO_GC | Das Attribut ist nicht zulässig, aus Sicherheitsgründen auf GC repliziert werden. |
| 8551 | 0x00002167 | ERROR_DS_NO_CHECKPOINT_WITH_PDC | Der Prüfpunkt mit dem PDC konnte nicht eingenommen werden, denn es gibt zu viele Änderungen derzeit verarbeitet wird. |
| 8552 | 0x00002168 | ERROR_DS_SOURCE_AUDITING_NOT_ENABLED | Der Vorgang erfordert die Quelle Domäne Überwachung aktiviert werden. |
| 8553 | 0x00002169 | ERROR_DS_CANT_CREATE_IN_NONDOMAIN_NC | Sicherheitsprinzipalobjekte können nur im Domänennamenkontext erstellt werden. |
| 8554 | 0x0000216A | ERROR_DS_INVALID_NAME_FOR_SPN | Ein Service Principal Name (SPN) konnte nicht erstellt werden, da der angegebene Hostname nicht im notwendigen Format ist. |
| 8555 | 0x0000216B | ERROR_DS_FILTER_USES_CONSTRUCTED_ATTRIBUTES | Ein Filter wurde übergeben, die konstruierten Attribute verwendet. |
| 8556 | 0x0000216C | ERROR_DS_UNICODEPWD_NOT_IN_QUOTES | Der Attributwert "unicodePWD" muss in Anführungszeichen eingeschlossen werden. |
| 8557 | 0x0000216D | ERROR_DS_MACHINE_ACCOUNT_QUOTA_EXCEEDED | Ihr Computer konnte der Domäne nicht hinzugefügt. Sie haben die maximale Anzahl der Computerkonten überschritten, die Sie berechtigt sind, in dieser Domäne zu erstellen. Wenden Sie sich an den Systemadministrator, um diese Grenze zurücksetzen oder erhöht haben. |
| 8558 | 0x0000216E | ERROR_DS_MUST_BE_RUN_ON_DST_DC | Aus Sicherheitsgründen muss der Vorgang auf der Ziel-DC ausgeführt werden. |
| 8559 | 0x0000216F | ERROR_DS_SRC_DC_MUST_BE_SP4_OR_GREATER | Aus Sicherheitsgründen muss der Quell-DC NT4SP4 sein oder größer. |
| 8560 | 0x00002170 | ERROR_DS_CANT_TREE_DELETE_CRITICAL_OBJ | Kritische Directory Service System-Objekte können während der Baum-Delete-Operationen gelöscht werden. Das Baum löschen kann teilweise durchgeführt wurden. |
| 8561 | 0x00002171 | ERROR_DS_INIT_FAILURE_CONSOLE | Verzeichnisdienste konnte aufgrund des folgenden Fehlers nicht gestartet: %1. Fehlerstatus: 0 x %2. Bitte klicken Sie auf OK zum Herunterfahren des Systems. Sie können die Wiederherstellungskonsole verwenden, um das System weiter zu diagnostizieren. |
| 8562 | 0x00002172 | ERROR_DS_SAM_INIT_FAILURE_CONSOLE | Sicherheitskontenverwaltung ist aufgrund des folgenden Fehlers fehlgeschlagen: %1. Fehlerstatus: 0 x %2. Bitte klicken Sie auf OK zum Herunterfahren des Systems. Sie können die Wiederherstellungskonsole verwenden, um das System weiter zu diagnostizieren. |
| 8563 | 0x00002173 | ERROR_DS_FOREST_VERSION_TOO_HIGH | Diese Version von Windows ist zu alt, um das aktuelle Verzeichnis Wald Verhalten zu unterstützen. Sie müssen das Betriebssystem auf diesem Server aktualisieren, bevor sie einen Domänencontroller in dieser Gesamtstruktur werden kann. |
| 8564 | 0x00002174 | ERROR_DS_DOMAIN_VERSION_TOO_HIGH | Diese Windows-Version ist zu alt, um das aktuelle Domain-Verhalten zu unterstützen. Sie müssen das Betriebssystem auf diesem Server aktualisieren, bevor sie einen Domänencontroller in dieser Domäne werden kann. |
| 8565 | 0x00002175 | ERROR_DS_FOREST_VERSION_TOO_LOW | Diese Windows-Version unterstützt nicht mehr die Verhaltensversion in diesem Directory-Gesamtstruktur verwendet. Sie müssen die Verhaltensversion Wald ausbauen, bevor dieser Server ein Domänencontroller in der Gesamtstruktur werden kann. |

| Fehler | | | Beschreibung |
|---------|-------------|---|--|
| Dezimal | Hexadezimal | Name | |
| 8566 | 0x00002176 | ERROR_DS_DOMAIN_VERSION_TOO_LOW | Diese Windows-Version unterstützt nicht mehr die Verhaltensversion in dieser Domäne verwendet. Sie müssen die Verhaltensversion der Domäne voraus, bevor dieser Server ein Domänencontroller in der Domäne werden kann. |
| 8567 | 0x00002177 | ERROR_DS_INCOMPATIBLE_VERSION | Der Windows-Version ist inkompatibel mit der Verhaltensversion der Domäne oder Gesamtstruktur. |
| 8568 | 0x00002178 | ERROR_DS_LOW_DSA_VERSION | Die Verhaltensversion kann nicht auf den angeforderten Wert erhöht werden, weil noch Domänencontroller mit Versionen kleiner als der angeforderte Wert vorhanden sind. |
| 8569 | 0x00002179 | ERROR_DS_NO_BEHAVIOR_VERSION_IN_MIXEDDOMAIN | Der Versionswert Verhalten kann nicht erhöht werden, während die Domäne noch im gemischten Domänenmodus befindet. Zuerst müssen Sie die Domäne in den einheitlichen Modus ändern, bevor die Verhaltensversion erhöht. |
| 8570 | 0x0000217A | ERROR_DS_NOT_SUPPORTED_SORT_ORDER | Die angeforderte Sortierreihenfolge wird nicht unterstützt. |
| 8571 | 0x0000217B | ERROR_DS_NAME_NOT_UNIQUE | Gefunden Sie ein Objekt mit nicht eindeutigen Namen. |
| 8572 | 0x0000217C | ERROR_DS_MACHINE_ACCOUNT_CREATED_PRENT4 | Das Computerkonto wurde vor NT4 erstellt. Das Konto muss neu erstellt werden. |
| 8573 | 0x0000217D | ERROR_DS_OUT_OF_VERSION_STORE | Die Datenbank ist außerhalb des Versionspeichers. |
| 8574 | 0x0000217E | ERROR_DS_INCOMPATIBLE_CONTROLS_USED | Nicht fortsetzen, weil mehrere widersprüchliche Steuerelemente verwendet wurden. |
| 8575 | 0x0000217F | ERROR_DS_NO_REF_DOMAIN | Keine gültige Sicherheitsbeschreibung-Referenzdomäne für diese Partition gefunden. |
| 8576 | 0x00002180 | ERROR_DS_RESERVED_LINK_ID | Schemaaktualisierung ist fehlgeschlagen: der Verknüpfungskennung ist reserviert. |
| 8577 | 0x00002181 | ERROR_DS_LINK_ID_NOT_AVAILABLE | Schemaaktualisierung ist fehlgeschlagen: Es gibt keine Link-Bezeichner. |
| 8578 | 0x00002182 | ERROR_DS_AG_CANT_HAVE_UNIVERSAL_MEMBER | Eine Kontogruppe kann keine universelle Gruppe als Mitglied haben. |
| 8579 | 0x00002183 | ERROR_DS_MODIFYDN_DISALLOWED_BY_INSTANCE_TYPE | Umbenennen Sie oder verschieben Sie, sind auf Benennung Kontext Köpfe oder schreibgeschützten Objekten nicht zulässig. |
| 8580 | 0x00002184 | ERROR_DS_NO_OBJECT_MOVE_IN_SCHEMA_NAMESPACE | Verschiebevorgänge sind auf Objekten im Schemanamenskontext nicht zulässig. |
| 8581 | 0x00002185 | ERROR_DS_MODIFYDN_DISALLOWED_BY_FLAG | Ein Systemflag festgelegt wurde, auf das Objekt und lässt nicht das Objekt verschoben oder umbenannt werden. |
| 8582 | 0x00002186 | ERROR_DS_MODIFYDN_WRONG_GRANDPARENT | Dieses Objekt ist nicht zulässig, dessen Großeltern-Container ändern. Bewegungen sind für dieses Objekt nicht verboten, aber beschränken sich auf Geschwister-Container. |
| 8583 | 0x00002187 | ERROR_DS_NAME_ERROR_TRUST_REFERRAL | Nicht vollständig auflösen, eine Überweisung in eine andere Gesamtstruktur erstellt wird. |
| 8584 | 0x00002188 | ERROR_NOT_SUPPORTED_ON_STANDARD_SERVER | Die angeforderte Aktion ist auf standard-Server nicht unterstützt. |
| 8585 | 0x00002189 | ERROR_DS_CANT_ACCESS_REMOTE_PART_OF_AD | Eine Partition von Active Directory befindet sich auf einem Remoteserver konnte nicht zugegriffen werden. Stellen Sie sicher, dass mindestens ein Server für die Partition ausgeführt wird, in Frage. |
| 8586 | 0x0000218A | ERROR_DS_CR_IMPOSSIBLE_TO_VALIDATE | Das Verzeichnis kann nicht den vorgeschlagenen Namen naming Context (oder Partition) überprüfen, denn es kein Replikat hält noch kann es ein Replikat des Namenskontextes oberhalb des vorgeschlagenen Namenskontextes kontaktieren. Bitte stellen Sie sicher, dass der übergeordnete Namenskontext richtig im DNS registriert ist und dass mindestens ein Replikat dieses Namenskontextes vom Domänennamensmaster erreichbar. |
| 8587 | 0x0000218B | ERROR_DS_THREAD_LIMIT_EXCEEDED | Der Thread für diese Anforderung wurde überschritten. |

| Fehler | | | Beschreibung |
|--------------|-------------|----------------------|--|
| Dezi-
mal | Hexadezimal | Name | |
| 8588 | 0x0000218C | ERROR_DS_NOT_CLOSEST | Der globale Katalogserver ist nicht im Schrank-
Standort. |

| | | Fehler | | Beschreibung |
|---------|-------------|--|--|--|
| Dezimal | Hexadezimal | Name | | |
| 9001 | 0x00002329 | DNS_ERROR_RCODE_FORMAT_ERROR | | DNS-Server nicht in der Lage, Format zu interpretieren. |
| 9002 | 0x0000232A | DNS_ERROR_RCODE_SERVER_FAILURE | | DNS-Server-Fehler. |
| 9003 | 0x0000232B | DNS_ERROR_RCODE_NAME_ERROR | | DNS-Name ist nicht vorhanden. |
| 9004 | 0x0000232C | DNS_ERROR_RCODE_NOT_IMPLEMENTED | | DNS-Anfrage vom Name-Server nicht unterstützt. |
| 9005 | 0x0000232D | DNS_ERROR_RCODE_REFUSED | | DNS Betrieb verweigert. |
| 9006 | 0x0000232E | DNS_ERROR_RCODE_YXDOMAIN | | DNS-Namen, die nicht existieren sollte, ist vorhanden. |
| 9007 | 0x0000232F | DNS_ERROR_RCODE_YXRRSET | | DNS-RR-Menge, die nicht vorhanden sein sollte, ist vorhanden. |
| 9008 | 0x00002330 | DNS_ERROR_RCODE_NXRRSET | | DNS-RR-Reihe, die vorhanden sind, ist nicht vorhanden. |
| 9009 | 0x00002331 | DNS_ERROR_RCODE_NOTAUTH | | DNS-Server nicht autorisierend für die Zone. |
| 9010 | 0x00002332 | DNS_ERROR_RCODE_NOTZONE | | DNS-Name im Update oder Prereq ist nicht in der Zone. |
| 9016 | 0x00002338 | DNS_ERROR_RCODE_BADSIG | | Die DNS-Signatur konnte nicht überprüft werden. |
| 9017 | 0x00002339 | DNS_ERROR_RCODE_BADKEY | | DNS-bad Taste. |
| 9018 | 0x0000233A | DNS_ERROR_RCODE_BADTIME | | Gültigkeit der DNS-Signatur ist abgelaufen. |
| 9501 | 0x0000251D | DNS_INFO_NO_RECORDS | | Keine Datensätze gefunden für angegebenen DNS-Abfrage. |
| 9502 | 0x0000251E | DNS_ERROR_BAD_PACKET | | Schlechten DNS-Paket. |
| 9503 | 0x0000251F | DNS_ERROR_NO_PACKET | | Kein DNS-Paket. |
| 9504 | 0x00002520 | DNS_ERROR_RCODE | | DNS-Fehler, Rcode überprüfen. |
| 9505 | 0x00002521 | DNS_ERROR_UNSECURE_PACKET | | Ungesicherte DNS-Paket. |
| 9551 | 0x0000254F | DNS_ERROR_INVALID_TYPE | | Ungültiger DNS-Typ. |
| 9552 | 0x00002550 | DNS_ERROR_INVALID_IP_ADDRESS | | Ungültige IP-Adresse. |
| 9553 | 0x00002551 | DNS_ERROR_INVALID_PROPERTY | | Ungültige Eigenschaft. |
| 9554 | 0x00002552 | DNS_ERROR_TRY_AGAIN_LATER | | Versuchen Sie DNS-Vorgang später noch einmal. |
| 9555 | 0x00002553 | DNS_ERROR_NOT_UNIQUE | | Eintrag für den angegebenen Namen und Typ ist nicht eindeutig. |
| 9556 | 0x00002554 | DNS_ERROR_NON_RFC_NAME | | DNS-Name entspricht nicht RFC-Spezifikationen. |
| 9557 | 0x00002555 | DNS_STATUS_FQDN | | DNS-Name ist ein vollständig qualifizierter DNS-Name. |
| 9558 | 0x00002556 | DNS_STATUS_DOTTED_NAME | | DNS-Name ist (Multilabel) punktiert. |
| 9559 | 0x00002557 | DNS_STATUS_SINGLE_PART_NAME | | DNS-Name ist eine einteilige. |
| 9560 | 0x00002558 | DNS_ERROR_INVALID_NAME_CHAR | | DSN-Name enthält ein ungültiges Zeichen. |
| 9561 | 0x00002559 | DNS_ERROR_NUMERIC_NAME | | DNS-Name ist vollständig numerisch. |
| 9601 | 0x00002581 | DNS_ERROR_ZONE_DOES_NOT_EXIST | | DNS-Zone ist nicht vorhanden. |
| 9602 | 0x00002582 | DNS_ERROR_NO_ZONE_INFO | | DNS-Zoneninformationen nicht verfügbar. |
| 9603 | 0x00002583 | DNS_ERROR_INVALID_ZONE_OPERATION | | Ungültiger Vorgang für die DNS-Zone. |
| 9604 | 0x00002584 | DNS_ERROR_ZONE_CONFIGURATION_ERROR | | Ungültige DNS-Zonenkonfiguration. |
| 9605 | 0x00002585 | DNS_ERROR_ZONE_HAS_NO_SOA_RECORD | | DNS-Zone ist kein Start des Autoritätsursprung (SOA)-Eintrags. |
| 9606 | 0x00002586 | DNS_ERROR_ZONE_HAS_NO_NS_RECORDS | | DNS-Zone besitzt keinen Namenservereintrag (NS). |
| 9607 | 0x00002587 | DNS_ERROR_ZONE_LOCKED | | DNS-Zone ist gesperrt. |
| 9608 | 0x00002588 | DNS_ERROR_ZONE_CREATION_FAILED | | Fehler beim DNS-Zone erstellen. |
| 9609 | 0x00002589 | DNS_ERROR_ZONE_ALREADY_EXISTS | | DNS-Zone ist bereits vorhanden. |
| 9610 | 0x0000258A | DNS_ERROR_AUTOZONE_ALREADY_EXISTS | | Automatische DNS-Zone ist bereits vorhanden. |
| 9611 | 0x0000258B | DNS_ERROR_INVALID_ZONE_TYPE | | Ungültiger DNS-Zonentyp. |
| 9612 | 0x0000258C | DNS_ERROR_SECONDARY_REQUIRES_MASTER_IP | | Sekundäre DNS-Zone erfordert master IP-Adresse. |
| 9613 | 0x0000258D | DNS_ERROR_ZONE_NOT_SECONDARY | | DNS-Zone ist nicht sekundär. |
| 9614 | 0x0000258E | DNS_ERROR_NEED_SECONDARY_ADDRESSES | | Benötigen Sie sekundäre IP-Adresse. |
| 9615 | 0x0000258F | DNS_ERROR_WINS_INIT_FAILED | | WINS-Initialisierung ist fehlgeschlagen. |
| 9616 | 0x00002590 | DNS_ERROR_NEED_WINS_SERVERS | | Benötigen Sie WINS-Server. |
| 9617 | 0x00002591 | DNS_ERROR_NBSTAT_INIT_FAILED | | NBSTAT-Initialisierungsaufwurf ist fehlgeschlagen. |
| 9618 | 0x00002592 | DNS_ERROR_SOA_DELETE_INVALID | | Ungültiges Löschen des Autoritätsursprung (SOA) |

| Fehler | | | Beschreibung |
|---------|-------------|-------------------------------------|--|
| Dezimal | Hexadezimal | Name | |
| 9619 | 0x00002593 | DNS_ERROR_FORWARDER_ALREADY_EXISTS | Eine bedingte Weiterleitung-Zone ist für diesen Namen bereits vorhanden. |
| 9651 | 0x000025B3 | DNS_ERROR_PRIMARY_REQUIRES_DATAFILE | Primäre DNS-Zone erfordert eine Datendatei. |
| 9652 | 0x000025B4 | DNS_ERROR_INVALID_DATAFILE_NAME | Ungültige Datafile Name für DNS-Zone. |
| 9653 | 0x000025B5 | DNS_ERROR_DATAFILE_OPEN_FAILURE | Öffnen Sie die Datendatei für die DNS-Zone konnte nicht. |
| 9654 | 0x000025B6 | DNS_ERROR_FILE_WRITEBACK_FAILED | Konnte die Datendatei für die DNS-Zone zu schreiben. |
| 9655 | 0x000025B7 | DNS_ERROR_DATAFILE_PARSING | Fehler beim Lesen der Datendatei für die DNS-Zone. |
| 9701 | 0x000025E5 | DNS_ERROR_RECORD_DOES_NOT_EXIST | DNS-Eintrag ist nicht vorhanden. |
| 9702 | 0x000025E6 | DNS_ERROR_RECORD_FORMAT | DNS-Datensatzformat Fehler. |
| 9703 | 0x000025E7 | DNS_ERROR_NODE_CREATION_FAILED | Knoten-Erstellung-Fehler im DNS. |
| 9704 | 0x000025E8 | DNS_ERROR_UNKNOWN_RECORD_TYPE | Unbekannten DNS-Datensatztyp. |
| 9705 | 0x000025E9 | DNS_ERROR_RECORD_TIMED_OUT | DNS-Eintrag Zeitüberschreitung. |
| 9706 | 0x000025EA | DNS_ERROR_NAME_NOT_IN_ZONE | Name nicht in der DNS-Zone. |
| 9707 | 0x000025EB | DNS_ERROR_CNAME_LOOP | CNAME-Schleife gefunden. |
| 9708 | 0x000025EC | DNS_ERROR_NODE_IS_CNAME | Knoten ist ein CNAME DNS-Eintrag. |
| 9709 | 0x000025ED | DNS_ERROR_CNAME_COLLISION | Ein CNAME-Eintrag für angegebenen Namen ist bereits vorhanden. |
| 9710 | 0x000025EE | DNS_ERROR_RECORD_ONLY_AT_ZONE_ROOT | Notieren Sie nur am DNS-Zonenstamm. |
| 9711 | 0x000025EF | DNS_ERROR_RECORD_ALREADY_EXISTS | DNS-Eintrag ist bereits vorhanden. |
| 9712 | 0x000025F0 | DNS_ERROR_SECONDARY_DATA | Sekundäre DNS-Zone Datenfehler. |
| 9713 | 0x000025F1 | DNS_ERROR_NO_CREATE_CACHE_DATA | DNS-Cachedaten konnten nicht erstellt werden. |
| 9714 | 0x000025F2 | DNS_ERROR_NAME_DOES_NOT_EXIST | DNS-Name ist nicht vorhanden. |
| 9715 | 0x000025F3 | DNS_WARNING_PTR_CREATE_FAILED | Zeigereintrag (PTR) konnte nicht erstellt werden. |
| 9716 | 0x000025F4 | DNS_WARNING_DOMAIN_UNDELETED | DNS-Domäne wurde wiederhergestellt. |
| 9717 | 0x000025F5 | DNS_ERROR_DS_UNAVAILABLE | Der Verzeichnisdienst ist nicht verfügbar. |
| 9718 | 0x000025F6 | DNS_ERROR_DS_ZONE_ALREADY_EXISTS | DNS-Zone ist bereits im Verzeichnisdienst vorhanden. |
| 9719 | 0x000025F7 | DNS_ERROR_NO_BOOTFILE_IF_DS_ZONE | DNS-Server nicht erstellen oder lesen die Startdatei für den Verzeichnisdienst integrierte DNS-Zone. |
| 9751 | 0x00002617 | DNS_INFO_AXFR_COMPLETE | DNS-AXFR (Zonenübertragung) abgeschlossen. |
| 9752 | 0x00002618 | DNS_ERROR_AXFR | DNS-Zonenübertragung ist fehlgeschlagen. |
| 9753 | 0x00002619 | DNS_INFO_ADDED_LOCAL_WINS | Zusätzlichen lokalen WINS-Server. |
| 9801 | 0x00002649 | DNS_STATUS_CONTINUE_NEEDED | Sichere Updates Aufruf muss Aktualisierungsanforderung fortsetzen. |
| 9851 | 0x0000267B | DNS_ERROR_NO_TCPIP | Netzwerkprotokoll TCP/IP nicht installiert |
| 9852 | 0x0000267C | DNS_ERROR_NO_DNS_SERVERS | Keine DNS-Server für das lokale System konfiguriert. |
| 9901 | 0x000026AD | DNS_ERROR_DP_DOES_NOT_EXIST | Die angegebene Verzeichnispertition existiert nicht. |
| 9902 | 0x000026AE | DNS_ERROR_DP_ALREADY_EXISTS | Die angegebene Verzeichnispertition ist bereits vorhanden. |
| 9903 | 0x000026AF | DNS_ERROR_DP_NOT_ENLISTED | Der DS ist nicht in der angegebenen Verzeichnispertition eingetragen. |
| 9904 | 0x000026B0 | DNS_ERROR_DP_ALREADY_ENLISTED | Der DS wird in die angegebene Verzeichnispertition bereits eingetragen. |

| Fehler | | | Beschreibung |
|---------|-------------|--------------------|---|
| Dezimal | Hexadezimal | Name | |
| 10004 | 0x00002714 | WSAEINTR | Ein Blockierungsvorgang wurde durch einen Aufruf von WSACancelBlockingCall unterbrochen. |
| 10009 | 0x00002719 | WSAEBADF | Gelieferten Datei-Handle ist ungültig. |
| 10013 | 0x0000271D | WSAEACCES | Es wurde versucht, auf einen Socket in einer Weise, die durch die Zugriffsberechtigungen verboten. |
| 10014 | 0x0000271E | WSAEFAULT | Das System entdeckt eine ungültige Zeigeradresse bei dem Versuch, das Zeigerargument in einem Aufruf verwenden. |
| 10022 | 0x00002726 | WSAEINVAL | Ein ungültiges Argument wurde angegeben. |
| 10024 | 0x00002728 | WSAEMFILE | Zu viele geöffnete Sockets. |
| 10035 | 0x00002733 | WSAEWOULDBLOCK | Ein nicht blockierender Socketvorgang konnte nicht sofort abgeschlossen werden. |
| 10036 | 0x00002734 | WSAEINPROGRESS | Ein Blockierungsvorgang wird momentan ausgeführt. |
| 10037 | 0x00002735 | WSAEALREADY | Ein Vorgang bezog sich auf einem nicht blockierenden Socket, die bereits einen Vorgang ausgeführt. |
| 10038 | 0x00002736 | WSAENOTSOCK | Ein Vorgang bezog sich auf etwas, das kein Socket ist. |
| 10039 | 0x00002737 | WSAEDESTADDRREQ | Eine erforderliche Adresse wurde aus einem Vorgang auf einem Socket weggelassen. |
| 10040 | 0x00002738 | WSAEMSGSIZE | Eine Meldung, die über einen Datagrammsocket gesendet wurde größer als den internen Nachrichtenpuffer oder einige andere Netzwerk-Limit, oder der Puffer verwendet, um ein Datagramm zu empfangen war kleiner als das Datagramm selbst. |
| 10041 | 0x00002739 | WSAEPROTOTYPE | Im Funktionsaufruf Socket wurde ein Protokoll angegeben, das Semantik des angeforderten Sockettyps nicht unterstützt. |
| 10042 | 0x0000273A | WSAENOPROTOOPT | In einem Aufruf von Getsockopt oder Setsockopt wurde eine nicht unterstützte Option bzw. Ebene angegeben. |
| 10043 | 0x0000273B | WSAEPROTONOSUPPORT | Das angeforderte Protokoll wurde nicht in das System konfiguriert, oder keine Implementierung dafür vorhanden. |
| 10044 | 0x0000273C | WSAESOCKTOSUPPORT | Die Unterstützung für den angegebenen Sockettyp existiert nicht in dieser Adressfamilie. |
| 10045 | 0x0000273D | WSAEOPNOTSUPP | Der Vorgang wird nicht unterstützt für den Typ des Objekts, auf die verwiesen wird. |
| 10046 | 0x0000273E | WSAEPFNOSUPPORT | Die Protokollfamilie wurde nicht in das System konfiguriert oder keine Implementierung dafür vorhanden. |
| 10047 | 0x0000273F | WSAEAFNOSUPPORT | Eine Adresse, die nicht kompatibel mit dem angeforderten Protokoll wurde verwendet. |
| 10048 | 0x00002740 | WSAEADDRINUSE | Regel nicht nur eine Verwendung von jede Socketadresse (Protokoll-Netzwerk-Adresse/ Anschluss) ist zulässig. |
| 10049 | 0x00002741 | WSAEADDRNOTAVAIL | Die angeforderte Adresse ist in seinem Kontext ungültig. |
| 10050 | 0x00002742 | WSAENETDOWN | Bei einem Socketvorgang ist ein tot Netzwerk aufgetreten. |
| 10051 | 0x00002743 | WSAENETUNREACH | Ein Socketvorgang bezog sich auf ein Netzwerk nicht erreichbar. |
| 10052 | 0x00002744 | WSAENETRESET | Die Verbindung wurde aufgrund von KeepAlive-Aktivitäten erkennen eines Fehlers während die Operation im Gange war gebrochen. |
| 10053 | 0x00002745 | WSAECONNABORTED | Eine bestehende Verbindung wurde von der Software in Ihrem Hostcomputer abgebrochen. |
| 10054 | 0x00002746 | WSAECONNRESET | Eine vorhandene Verbindung wurde vom Remotehost geschlossen. |
| 10055 | 0x00002747 | WSAENOBUFS | Ein Vorgang auf einem Socket konnte nicht ausgeführt werden, weil das System ausreichend Pufferspeicher fehlte oder eine Warteschlange voll war. |

| Fehler | | | Beschreibung |
|---------|-------------|------------------------|--|
| Dezimal | Hexadezimal | Name | |
| 10056 | 0x00002748 | WSAEISCONN | Eine Verbindungsanforderung wurde auf einen bereits verbundenen Socket. |
| 10057 | 0x00002749 | WSAENOTCONN | Eine Anforderung zum Senden oder empfangen von Daten wurde verhindert, da der Socket nicht verbunden ist und (beim Senden über einen Datagrammssocket mit einem Sendto-Aufruf) keine Adresse angegeben wurde. |
| 10058 | 0x0000274A | WSAESHUTDOWN | Eine Anforderung zum Senden oder empfangen von Daten wurde verhindert, da der Socket hatte bereits in diese Richtung mit einem vorangegangenen Shutdown-Aufruf heruntergefahren worden. |
| 10059 | 0x0000274B | WSAETOOMANYREFS | Zu viele Verweise auf einige Kernelobjekt. |
| 10060 | 0x0000274C | WSAETIMEDOUT | Ein Verbindungsversuch ist fehlgeschlagen, da die verbundene Partei nicht richtig geantwortet hat, nach einer gewissen Zeit, oder die hergestellte Verbindung ist fehlgeschlagen, da der verbundene Host nicht reagiert hat. |
| 10061 | 0x0000274D | WSAECONNREFUSED | Keine Verbindung konnte hergestellt werden, da der Zielcomputer aktiv es verweigert. |
| 10062 | 0x0000274E | WSAELOOP | Name kann nicht übersetzt werden. |
| 10063 | 0x0000274F | WSAENAMETOOLONG | Namenskomponente oder Name war zu lang. |
| 10064 | 0x00002750 | WSAEHOSTDOWN | Bei einem Socketvorgang ist fehlgeschlagen, da der Zielhost gedrückt wurde. |
| 10065 | 0x00002751 | WSAEHOSTUNREACH | Ein Socketvorgang bezog sich auf einen Host nicht erreichbar. |
| 10066 | 0x00002752 | WSAENOTEMPTY | Ein Verzeichnis kann nicht entfernt werden, die nicht leer ist. |
| 10067 | 0x00002753 | WSAEPROCLIM | Eine Windows Sockets-Implementation kann ein Limit für die Anzahl der Anwendungen bestehen, die zugleich auf diese zugreifen kann. |
| 10068 | 0x00002754 | WSAEUSERS | Lief der Quote. |
| 10069 | 0x00002755 | WSAEDQUOT | Datenträgerkontingent lief. |
| 10070 | 0x00002756 | WSAESTALE | Datei-Handle-Verweis ist nicht mehr verfügbar. |
| 10071 | 0x00002757 | WSAEREMOTE | Element ist nicht lokal verfügbar. |
| 10091 | 0x0000276B | WSASYSNOTREADY | WSAStartup funktioniert Funktion nicht zu dieser Zeit, da das zugrunde liegende System verwendeten Netzwerkdienste derzeit nicht verfügbar ist. |
| 10092 | 0x0000276C | WSAVERNOTSUPPORTED | Die angeforderte Windows Sockets-Version wird nicht unterstützt. |
| 10093 | 0x0000276D | WSANOTINITIALISED | Die Anwendung hat WSAStartup nicht aufgerufen, oder WSAStartup fehlgeschlagen. |
| 10101 | 0x00002775 | WSAEDISCON | Wird von WSARcv oder WSARcvFrom, um anzugeben, dass die Gegenseite eine ordnungsgemäßes Herunterfahren-Sequenz eingeleitet hat. |
| 10102 | 0x00002776 | WSAENOMORE | Von WSALookupServiceNext können keine weiteren Ergebnisse zurückgegeben werden. |
| 10103 | 0x00002777 | WSAECANCELLED | WSALookupServiceEnd wurde aufgerufen, während dieser Aufruf noch verarbeitet wurde. Der Aufruf wurde abgebrochen. |
| 1050° | 0x00002778 | WSAEINVALIDPROCTABLE | Die Prozedur-Aufruf-Tabelle ist ungültig. |
| 10105 | 0x00002779 | WSAEINVALIDPROVIDER | Der angeforderte Dienstanbieter ist ungültig. |
| 10106 | 0x0000277A | WSAEPROVIDERFAILEDINIT | Der angeforderte Dienstanbieter konnte nicht geladen oder initialisiert werden. |
| 10107 | 0x0000277B | WSASYSALLFAILURE | Ein Systemaufruf, der nie ausfällt ist fehlgeschlagen. |
| 10108 | 0x0000277C | WSASERVICE_NOT_FOUND | Keine solchen Dienst ist bekannt. Der Dienst kann nicht im angegebenen Namespace gefunden werden. |
| 10109 | 0x0000277D | WSATYPE_NOT_FOUND | Die angegebene Klasse wurde nicht gefunden. |
| 10110 | 0x0000277E | WSA_E_NO_MORE | Von WSALookupServiceNext können keine weiteren Ergebnisse zurückgegeben werden. |

| Fehler | | | Beschreibung |
|---------|-------------|----------------------------|--|
| Dezimal | Hexadezimal | Name | |
| 10111 | 0x0000277F | WSA_E_CANCELLED | WSALookupServiceEnd wurde aufgerufen, während dieser Aufruf noch verarbeitet wurde. Der Aufruf wurde abgebrochen. |
| 10112 | 0x00002780 | WSAEREFUSED | Eine Datenbankabfrage ist fehlgeschlagen, da es aktiv verweigert wurde. |
| 11001 | 0x00002AF9 | WSAHOST_NOT_FOUND | Keine solche Host ist unbekannt. |
| 11002 | 0x00002AFA | WSATRY_AGAIN | Dies ist normalerweise ein zeitweiliger Fehler bei der Auflösung von Hostnamen und bedeutet, dass der lokale Server keine Antwort aus einem autorisierenden Server erhalten. |
| 11003 | 0x00002AFB | WSANO_RECOVERY | Während eine Datenbanksuche ist ein nicht behebbarer Fehler aufgetreten. |
| 11004 | 0x00002AFC | WSANO_DATA | Der angeforderte Name ist gültig und wurde in der Datenbank gefunden, aber es verfügt nicht über die richtigen zugeordneten Daten, die für gelöst. |
| 11005 | 0x00002AFD | WSA_QOS_RECEIVERS | Mindestens eine Reserve ist angekommen. |
| 11006 | 0x00002AFE | WSA_QOS_SENDERS | Mindestens einen Pfad ist angekommen. |
| 11007 | 0x00002AFF | WSA_QOS_NO_SENDERS | Es gibt keine Absender. |
| 11008 | 0x00002B00 | WSA_QOS_NO_RECEIVERS | Es gibt keine Empfänger. |
| 11009 | 0x00002B01 | WSA_QOS_REQUEST_CONFIRMED | Reserve ist bestätigt worden. |
| 11010 | 0x00002B02 | WSA_QOS_ADMISSION_FAILURE | Fehler aufgrund mangelnder Ressourcen. |
| 11011 | 0x00002B03 | WSA_QOS_POLICY_FAILURE | Aus administrativen Gründen - schlechte Anmeldeinformationen abgelehnt. |
| 11012 | 0x00002B04 | WSA_QOS_BAD_STYLE | Unbekannte oder widersprüchlichen Stil. |
| 11013 | 0x00002B05 | WSA_QOS_BAD_OBJECT | Problem mit einen Teil des Puffers Filter- oder Providerspecific im Allgemeinen. |
| 11014 | 0x00002B06 | WSA_QOS_TRAFFIC_CTRL_ERROR | Problem mit einem Teil der Flowspec. |
| 11015 | 0x00002B07 | WSA_QOS_GENERIC_ERROR | General QOS-Fehler. |
| 11016 | 0x00002B08 | WSA_QOS_ESERVICETYPE | Eine ungültige oder unbekannte Dienstyp wurde der Flowspec gefunden. |
| 11017 | 0x00002B09 | WSA_QOS_EFLOWSPEC | In der QOS-Struktur wurde eine ungültige oder inkonsistente Flowspec gefunden. |
| 11018 | 0x00002B0A | WSA_QOS_EPROVSPECBUF | Ungültige QOS-spezifischen Anbieterpuffer. |
| 11019 | 0x00002B0B | WSA_QOS_EFILTERSTYLE | Ein ungültiger QOS-Filter-Stil wurde verwendet. |
| 11020 | 0x00002B0C | WSA_QOS_EFILTERTYPE | Ein ungültiger QOS-Filter-Typ verwendet wurde. |
| 11021 | 0x00002B0D | WSA_QOS_EFILTERCOUNT | Eine falsche Anzahl an QOS FILTERSPECS wurden in die FLOWDESCRIPTOR angegeben. |
| 11022 | 0x00002B0E | WSA_QOS_EOBJLENGTH | Im QOS-spezifischen Anbieterpuffer wurde ein Objekt mit einem ungültigen ObjectLength-Feld angegeben. |
| 11023 | 0x00002B0F | WSA_QOS_EFLOWCOUNT | In der QOS-Struktur wurde eine falsche Anzahl von Flow-Deskriptoren angegeben. |
| 11024 | 0x00002B10 | WSA_QOS_EUNKNOWNPSOBJ | Im QOS-spezifischen Anbieterpuffer wurde ein unbekanntes Objekt gefunden. |
| 11025 | 0x00002B11 | WSA_QOS_EPOLICYOBJ | Im QOS-spezifischen Anbieterpuffer wurde ein ungültiges Richtlinienobjekt gefunden. |
| 11026 | 0x00002B12 | WSA_QOS_EFLOWDESC | Ein ungültiger QOS-Flow-Deskriptor wurde in die Liste der Flow-Deskriptoren gefunden. |
| 11027 | 0x00002B13 | WSA_QOS_EPSFLOWSPEC | Im QOS-spezifischen Anbieterpuffer wurde eine ungültige oder inkonsistente Flowspec gefunden. |
| 11028 | 0x00002B14 | WSA_QOS_EPSFILTERSPEC | Im QOS-spezifischen Anbieterpuffer wurde ein ungültiger FILTERSPEC gefunden. |
| 11029 | 0x00002B15 | WSA_QOS_ESDMODEOBJ | Im QOS-spezifischen Anbieterpuffer wurde eine ungültige Formobjekt verwerfen Modus gefunden. |
| 11030 | 0x00002B16 | WSA_QOS_ESHAPERATEOBJ | Im QOS-spezifischen Anbieterpuffer wurde ein ungültiges Strukturierungsratenobjekt gefunden. |
| 11031 | 0x00002B17 | WSA_QOS_RESERVED_PETYPE | Im QOS-spezifischen Anbieterpuffer wurde ein reserviertes Richtlinielement gefunden. |

| Fehler | | | Beschreibung |
|---------|-------------|--|--|
| Dezimal | Hexadezimal | Name | |
| 12000 | 0x00002EE0 | ERROR_SXS_SECTION_NOT_FOUND | Der angeforderte Abschnitt war nicht anwesend in den Aktivierungskontext. |
| 12001 | 0x00002EE1 | ERROR_SXS_CANT_GEN_ACTCTX | Diese Anwendung konnte nicht gestartet werden, weil die Anwendungskonfiguration falsch ist. Durch eine Neuinstallation der Anwendung könnte das Problem beheben. |
| 12002 | 0x00002EE2 | ERROR_SXS_INVALID_ACTCTXDATA_FORMAT | Das Anwendungsformat Datenbindung ist ungültig. |
| 12003 | 0x00002EE3 | ERROR_SXS_ASSEMBLY_NOT_FOUND | Die referenzierte Assemblierung ist nicht auf Ihrem System installiert. |
| 12004 | 0x00002EE4 | ERROR_SXS_MANIFEST_FORMAT_ERROR | Die manifest-Datei beginnt nicht mit der erforderlichen Tag und Format. |
| 12005 | 0x00002EE5 | ERROR_SXS_MANIFEST_PARSE_ERROR | Die manifest-Datei enthält einen oder mehrere Syntaxfehler. |
| 12006 | 0x00002EE6 | ERROR_SXS_ACTIVATION_CONTEXT_DISABLED | Die Anwendung hat versucht, einen deaktivierten Aktivierungskontext zu aktivieren. |
| 12007 | 0x00002EE7 | ERROR_SXS_KEY_NOT_FOUND | Der angeforderte Nachschlüssel wurde nicht in jedem aktiven Aktivierungskontext gefunden. |
| 12008 | 0x00002EE8 | ERROR_SXS_VERSION_CONFLICT | Eine Komponentenversion benötigt, die durch die Anwendungskonflikte mit einer anderen bereits aktiven Komponentenversion. |
| 12009 | 0x00002EE9 | ERROR_SXS_WRONG_SECTION_TYPE | Der Typ angefordert, Aktivierung Kontext Abschnitt nicht die verwendeten API-Abfrage entspricht. |
| 12010 | 0x00002EEA | ERROR_SXS_THREAD_QUERIES_DISABLED | Mangelnde Systemressourcen erforderte isolierten Aktivierung für den aktuellen Ausführungsthread an deaktiviert werden. |
| 12011 | 0x00002EEB | ERROR_SXS_PROCESS_DEFAULT_ALREADY_SET | Beim Versuch, den Prozess-Standard-Aktivierungskontext festzulegen ist fehlgeschlagen, da der Prozess-Standard-Aktivierung-Kontext bereits festgelegt wurde. |
| 12012 | 0x00002EEC | ERROR_SXS_UNKNOWN_ENCODING_GROUP | Bezeichner der Codierung angegeben wird nicht erkannt. |
| 12013 | 0x00002EED | ERROR_SXS_UNKNOWN_ENCODING | Die Kodierung angefordert wird nicht erkannt. |
| 12014 | 0x00002EEE | ERROR_SXS_INVALID_XML_NAMESPACE_URI | Das Manifest enthält einen Verweis auf einen ungültigen URI. |
| 12015 | 0x00002EEF | ERROR_SXS_ROOT_MANIFEST_DEPENDENCY_NOT_INSTALLED | Das Anwendungsmanifest enthält einen Verweis auf eine abhängige Assembly, die nicht installiert ist. |
| 12016 | 0x00002EF0 | ERROR_SXS_LEAF_MANIFEST_DEPENDENCY_NOT_INSTALLED | Das Manifest für eine Assembly, die von der Anwendung verwendeten hat einen Verweis auf eine abhängige Assembly, die nicht installiert ist. |
| 12017 | 0x00002EF1 | ERROR_SXS_INVALID_ASSEMBLY_IDENTITY_ATTRIBUTE | Das Manifest enthält ein Attribut für die Identität der Assembly, die nicht gültig ist. |
| 12018 | 0x00002EF2 | ERROR_SXS_MANIFEST_MISSING_REQUIRED_DEFAULT_NAMESPACE | Das Manifest fehlt die erforderliche Standardspezifikation Namespace auf das Assembly-Element. |
| 12019 | 0x00002EF3 | ERROR_SXS_MANIFEST_INVALID_REQUIRED_DEFAULT_NAMESPACE | Das Manifest hat einen Standardnamespace auf Assembly-Elements angegeben, aber sein Wert ist nicht "Urn: Schemas-Microsoft-com". |
| 12020 | 0x00002EF4 | ERROR_SXS_PRIVATE_MANIFEST_CROSS_PATH_WITH_REPARSE_POINT | Die private Manifeste Sonde hat den Analysepunkte Punkt-assoziierten Weg gekreuzt. |
| 12021 | 0x00002EF5 | ERROR_SXS_DUPLICATE_DLL_NAME | Zwei oder mehr Komponenten, vom Anwendungsmanifest direkt oder indirekt referenzierte haben Dateien mit demselben Namen. |
| 12022 | 0x00002EF6 | ERROR_SXS_DUPLICATE_WINDOWCLASS_NAME | Zwei oder mehr Komponenten, vom Anwendungsmanifest direkt oder indirekt referenzierte haben Fensterklassen mit demselben Namen. |
| 12023 | 0x00002EF7 | ERROR_SXS_DUPLICATE_CLSID | Zwei oder mehr Komponenten verwiesen direkt oder indirekt vom Anwendungsmanifest haben den gleichen COM-Server CLSIDs. |

| Fehler | | | Beschreibung |
|---------|-------------|--|--|
| Dezimal | Hexadezimal | Name | |
| 12024 | 0x00002EF8 | ERROR_SXS_DUPLICATE_IID | Zwei oder mehr Komponenten, vom Anwendungsmanifest direkt oder indirekt referenzierte haben Proxies für die gleichen IIDs der COM-Schnittstelle. |
| 12025 | 0x00002EF9 | ERROR_SXS_DUPLICATE_TLBID | Zwei oder mehr Komponenten, vom Anwendungsmanifest direkt oder indirekt referenzierte haben den gleichen COM-Typ-Bibliothek TLBIDs. |
| 12026 | 0x00002EFA | ERROR_SXS_DUPLICATE_PROGID | Zwei oder mehr Komponenten verwiesen direkt oder indirekt vom Anwendungsmanifest haben den gleichen COM-ProgIDs. |
| 12027 | 0x00002EFB | ERROR_SXS_DUPLICATE_ASSEMBLY_NAME | Zwei oder mehr Komponenten, vom Anwendungsmanifest direkt oder indirekt referenzierte sind verschiedene Versionen derselben Komponente nicht zulässig ist. |
| 12028 | 0x00002EFC | ERROR_SXS_FILE_HASH_MISMATCH | Eine Komponentendatei stimmt nicht mit den Überprüfungsinformationen im Komponentenmanifest überein. |
| 12029 | 0x00002EFD | ERROR_SXS_POLICY_PARSE_ERROR | Das Politik-Manifest enthält einen oder mehrere Syntaxfehler. |
| 12030 | 0x00002EFE | ERROR_SXS_XML_E_MISSINGQUOTE | Manifestverarbeitungsfehler: Ein Zeichenfolgenliteral wurde erwartet, aber keine Öffnung Anführungsstrich Zeichen wurde gefunden. |
| 12031 | 0x00002EFF | ERROR_SXS_XML_E_COMMENTSYNTAX | Manifestverarbeitungsfehler: Falsche Syntax wurde in einem Kommentar verwendet. |
| 12032 | 0x00002F00 | ERROR_SXS_XML_E_BADSTARTNAMECHAR | Manifestverarbeitungsfehler: Ein Name wurde mit einem ungültigen Zeichen gestartet. |
| 12033 | 0x00002F01 | ERROR_SXS_XML_E_BADNAMECHAR | Manifestverarbeitungsfehler: Ein Name enthält ein ungültiges Zeichen. |
| 12034 | 0x00002F02 | ERROR_SXS_XML_E_BADCHARINSTRING | Manifestverarbeitungsfehler: Ein String-literal enthält ein ungültiges Zeichen. |
| 12035 | 0x00002F03 | ERROR_SXS_XML_E_XMLDECLSYNTAX | Manifestverarbeitungsfehler: Ungültige Syntax für eine XML-Deklaration. |
| 12036 | 0x00002F04 | ERROR_SXS_XML_E_BADCHARDATA | Manifestverarbeitungsfehler: Ein ungültiges Zeichen wurde im Textinhalt gefunden. |
| 12037 | 0x00002F05 | ERROR_SXS_XML_E_MISSINGWHITESPACE | Manifestverarbeitungsfehler: Erforderliche Leerraum fehlte. |
| 12038 | 0x00002F06 | ERROR_SXS_XML_E_EXPECTINGTAGEND | Manifestverarbeitungsfehler: Das Zeichen '>' wurde erwartet. |
| 12039 | 0x00002F07 | ERROR_SXS_XML_E_MISSINGSEMICOLON | Manifestverarbeitungsfehler: Es wurde ein Semikolon erwartet. |
| 12040 | 0x00002F08 | ERROR_SXS_XML_E_UNBALANCEDPAREN | Manifestverarbeitungsfehler: Unausgeglichene Klammern. |
| 12041 | 0x00002F09 | ERROR_SXS_XML_E_INTERNALERROR | Manifestverarbeitungsfehler: Interner Fehler. |
| 12042 | 0x00002F0A | ERROR_SXS_XML_E_UNEXPECTED_WHITESPACE | Manifestverarbeitungsfehler: Leerraum ist an dieser Stelle nicht zulässig. |
| 12043 | 0x00002F0B | ERROR_SXS_XML_E_INCOMPLETE_ENCODING | Manifestverarbeitungsfehler: Ende der Datei erreicht ungültigen Status für die aktuelle Codierung. |
| 12044 | 0x00002F0C | ERROR_SXS_XML_E_MISSING_PAREN | Manifestverarbeitungsfehler: Fehlende Klammer. |
| 12045 | 0x00002F0D | ERROR_SXS_XML_E_EXPECTINGCLOSEQUOTE | Manifestverarbeitungsfehler: Ein Einzel- oder Doppelzitat schließendes Anführungszeichen Zeichen (' oder ') fehlt. |
| 12046 | 0x00002F0E | ERROR_SXS_XML_E_MULTIPLE_COLONS | Manifestverarbeitungsfehler: Mehrere Doppelpunkte dürfen nicht in einen Namen. |
| 12047 | 0x00002F0F | ERROR_SXS_XML_E_INVALID_DECIMAL | Manifestverarbeitungsfehler: Ungültiges Zeichen für Dezimalziffer. |
| 12048 | 0x00002F10 | ERROR_SXS_XML_E_INVALID_HEXIDECIMAL | Manifestverarbeitungsfehler: Ungültiges Zeichen für die Hexadezimalziffer. |
| 12049 | 0x00002F11 | ERROR_SXS_XML_E_INVALID_UNICODE | Manifestverarbeitungsfehler: Ungültige Unicode-Zeichenwert für diese Plattform. |
| 12050 | 0x00002F12 | ERROR_SXS_XML_E_WHITESPACEORQUESTIONMARK | Manifestverarbeitungsfehler: Erwartet Leerraum oder '?'. |
| 12051 | 0x00002F13 | ERROR_SXS_XML_E_UNEXPECTEDENDTAG | Manifestverarbeitungsfehler: End-Tag wurde an dieser Stelle nicht erwartet. |

| Fehler | | | Beschreibung |
|---------|-------------|---------------------------------------|---|
| Dezimal | Hexadezimal | Name | |
| 12052 | 0x00002F14 | ERROR_SXS_XML_E_UNCLOSEDTAG | Manifestverarbeitungsfehler: Die folgenden Tags waren nicht geschlossen: %1. |
| 12053 | 0x00002F15 | ERROR_SXS_XML_E_DUPLICATEATTRIBUTE | Manifestverarbeitungsfehler: Doppeltes Attribut. |
| 12054 | 0x00002F16 | ERROR_SXS_XML_E_MULTIPLEROOTS | Manifestverarbeitungsfehler: Einziger Top-Level-Element darf in einem XML-Dokument. |
| 12055 | 0x00002F17 | ERROR_SXS_XML_E_INVALIDATROOTLEVEL | Manifestverarbeitungsfehler: Ungültig auf der obersten Ebene des Dokuments. |
| 12056 | 0x00002F18 | ERROR_SXS_XML_E_BADXMLDECL | Manifestverarbeitungsfehler: Ungültige XML-Deklaration. |
| 12057 | 0x00002F19 | ERROR_SXS_XML_E_MISSINGROOT | Manifestverarbeitungsfehler: XML-Dokument muss ein Element der obersten Ebene. |
| 12058 | 0x00002F1A | ERROR_SXS_XML_E_UNEXPECTEDEOF | Manifestverarbeitungsfehler: Unerwartetes Dateieinde. |
| 12059 | 0x00002F1B | ERROR_SXS_XML_E_BADPEREFINSUBSET | Manifestverarbeitungsfehler: Parameter-Entitäten können nicht innerhalb von Markupdeklarationen in einer internen Teilmenge verwendet werden. |
| 12060 | 0x00002F1C | ERROR_SXS_XML_E_UNCLOSEDSTARTTAG | Manifestverarbeitungsfehler: Element wurde nicht geschlossen. |
| 12061 | 0x00002F1D | ERROR_SXS_XML_E_UNCLOSEDENDTAG | Manifestverarbeitungsfehler: Endelement fehlte das Zeichen '>'. |
| 12062 | 0x00002F1E | ERROR_SXS_XML_E_UNCLOSEDSTRING | Manifestverarbeitungsfehler: Ein Zeichenfolgenliteral wurde nicht geschlossen. |
| 12063 | 0x00002F1F | ERROR_SXS_XML_E_UNCLOSEDCOMMENT | Manifestverarbeitungsfehler: Ein Kommentar wurde nicht geschlossen. |
| 12064 | 0x00002F20 | ERROR_SXS_XML_E_UNCLOSEDDECL | Manifestverarbeitungsfehler: Eine Erklärung wurde nicht geschlossen. |
| 12065 | 0x00002F21 | ERROR_SXS_XML_E_UNCLOSEDCDATA | Manifestverarbeitungsfehler: Ein CDATA-Abschnitt wurde nicht geschlossen. |
| 12066 | 0x00002F22 | ERROR_SXS_XML_E_RESERVEDNAMESPACE | Manifestverarbeitungsfehler: Das Namespacepräfix darf nicht mit der reservierten Zeichenkette "Xml" beginnen. |
| 12067 | 0x00002F23 | ERROR_SXS_XML_E_INVALIDENCODING | Manifestverarbeitungsfehler: System der angegebenen Codierung unterstützt nicht. |
| 12068 | 0x00002F24 | ERROR_SXS_XML_E_INVALIDSWITCH | Manifestverarbeitungsfehler: Wechsel von der aktuellen Codierung auf angegebene Codierung nicht unterstützt. |
| 12069 | 0x00002F25 | ERROR_SXS_XML_E_BADXMLCASE | Manifestverarbeitungsfehler: Der Name "Xml" ist reserviert und muss klein geschrieben sein. |
| 12070 | 0x00002F26 | ERROR_SXS_XML_E_INVALID_STANDALONE | Manifestverarbeitungsfehler: Das Standalone-Attribut muss den Wert "Ja" oder "Nein". |
| 12071 | 0x00002F27 | ERROR_SXS_XML_E_UNEXPECTED_STANDALONE | Manifestverarbeitungsfehler: Das Standalone-Attribut kann nicht in externe Entitäten verwendet werden. |
| 12072 | 0x00002F28 | ERROR_SXS_XML_E_INVALID_VERSION | Manifestverarbeitungsfehler: Ungültige Versionsnummer. |
| 12073 | 0x00002F29 | ERROR_SXS_XML_E_MISSINGEQUALS | Manifestverarbeitungsfehler: Fehlendes Gleichheitszeichen zwischen und Attributwert. |
| 13000 | 0x000032C8 | ERROR_IPSEC_QM_POLICY_EXISTS | Die angegebene Schnellmodusrichtlinie ist bereits vorhanden. |
| 13001 | 0x000032C9 | ERROR_IPSEC_QM_POLICY_NOT_FOUND | Die angegebene Schnellmodusrichtlinie wurde nicht gefunden. |
| 13002 | 0x000032CA | ERROR_IPSEC_QM_POLICY_IN_USE | Die angegebene Schnellmodusrichtlinie wird verwendet. |
| 13003 | 0x000032CB | ERROR_IPSEC_MM_POLICY_EXISTS | Die angegebene Hauptmodusrichtlinie ist bereits vorhanden. |
| 13004 | 0x000032CC | ERROR_IPSEC_MM_POLICY_NOT_FOUND | Die angegebene Hauptmodusrichtlinie wurde nicht gefunden. |
| 13005 | 0x000032CD | ERROR_IPSEC_MM_POLICY_IN_USE | Die angegebene Hauptmodusrichtlinie wird verwendet. |
| 13006 | 0x000032CE | ERROR_IPSEC_MM_FILTER_EXISTS | Der angegebene Hauptmodusfilter ist bereits vorhanden. |
| 13007 | 0x000032CF | ERROR_IPSEC_MM_FILTER_NOT_FOUND | Die angegebene Hauptmodusfilter wurde nicht gefunden. |
| 13008 | 0x000032D0 | ERROR_IPSEC_TRANSPORT_FILTER_EXISTS | Der angegebenen Transport-Mode-Filter ist bereits vorhanden. |

| Fehler | | | Beschreibung |
|---------|-------------|---|--|
| Dezimal | Hexadezimal | Name | |
| 13009 | 0x000032D1 | ERROR_IPSEC_TRANSPORT_FILTER_NOT_FOUND | Der angegebenen Transport-Mode-Filter ist nicht vorhanden. |
| 13010 | 0x000032D2 | ERROR_IPSEC_MM_AUTH_EXISTS | Die angegebene Hauptmodus-Authentifizierungsliste ist vorhanden. |
| 13011 | 0x000032D3 | ERROR_IPSEC_MM_AUTH_NOT_FOUND | Die angegebene Hauptmodus-Authentifizierungsliste wurde nicht gefunden. |
| 13012 | 0x000032D4 | ERROR_IPSEC_MM_AUTH_IN_USE | Die angegebene Schnellmodusrichtlinie wird verwendet. |
| 13013 | 0x000032D5 | ERROR_IPSEC_DEFAULT_MM_POLICY_NOT_FOUND | Die angegebene Hauptmodusrichtlinie wurde nicht gefunden. |
| 13014 | 0x000032D6 | ERROR_IPSEC_DEFAULT_MM_AUTH_NOT_FOUND | Die angegebene Schnellmodusrichtlinie wurde nicht gefunden. |
| 13015 | 0x000032D7 | ERROR_IPSEC_DEFAULT_QM_POLICY_NOT_FOUND | Die manifest-Datei enthält einen oder mehrere Syntaxfehler. |
| 13016 | 0x000032D8 | ERROR_IPSEC_TUNNEL_FILTER_EXISTS | Die Anwendung hat versucht, einen deaktivierten Aktivierungskontext zu aktivieren. |
| 13017 | 0x000032D9 | ERROR_IPSEC_TUNNEL_FILTER_NOT_FOUND | Der angeforderte Nachschlageschlüssel wurde nicht in jedem aktiven Aktivierungskontext gefunden. |
| 13018 | 0x000032DA | ERROR_IPSEC_MM_FILTER_PENDING_DELETION | Der Hauptmodus-Filter steht zum Löschen. |
| 13019 | 0x000032DB | ERROR_IPSEC_TRANSPORT_FILTER_PENDING_DELETION | Der Transportfilter steht zum Löschen. |
| 13020 | 0x000032DC | ERROR_IPSEC_TUNNEL_FILTER_PENDING_DELETION | Der Tunnel-Filter steht zum Löschen. |
| 13021 | 0x000032DD | ERROR_IPSEC_MM_POLICY_PENDING_DELETION | Die Hauptmodus-Politik steht zum Löschen. |
| 13022 | 0x000032DE | ERROR_IPSEC_MM_AUTH_PENDING_DELETION | Das Hauptmodus-Authentifizierung-Bundle steht zum Löschen. |
| 13023 | 0x000032DF | ERROR_IPSEC_QM_POLICY_PENDING_DELETION | Die Schnellmodusrichtlinie steht zum Löschen. |
| 13801 | 0x000035E9 | ERROR_IPSEC_IKE_AUTH_FAIL | IKE-Authentifizierung-Anmeldeinformationen sind inakzeptabel. |
| 13802 | 0x000035EA | ERROR_IPSEC_IKE_ATTRIB_FAIL | IKE-Sicherheitsattribute sind inakzeptabel. |
| 13803 | 0x000035EB | ERROR_IPSEC_IKE_NEGOTIATION_PENDING | IKE-Aushandlung im Gange. |
| 13804 | 0x000035EC | ERROR_IPSEC_IKE_GENERAL_PROCESSING_ERROR | Allgemeine Verarbeitungsfehler. |
| 13805 | 0x000035ED | ERROR_IPSEC_IKE_TIMED_OUT | Verhandlung Zeitüberschreitung. |
| 13806 | 0x000035EE | ERROR_IPSEC_IKE_NO_CERT | IKE wurde kein gültiges Zertifikat gefunden. |
| 13807 | 0x000035EF | ERROR_IPSEC_IKE_SA_DELETED | IKE-Sicherheitszuordnung gelöscht von Peer vor Gründung abgeschlossen. |
| 13808 | 0x000035F0 | ERROR_IPSEC_IKE_SA_REAPED | IKE-Sicherheitszuordnung gelöscht vor Gründung abgeschlossen. |
| 13809 | 0x000035F1 | ERROR_IPSEC_IKE_MM_ACQUIRE_DROP | Aushandlungsanforderung saß in der Warteschlange zu lang. |
| 13810 | 0x000035F2 | ERROR_IPSEC_IKE_QM_ACQUIRE_DROP | Aushandlungsanforderung saß in der Warteschlange zu lang. |
| 13811 | 0x000035F3 | ERROR_IPSEC_IKE_QUEUE_DROP_MM | Aushandlungsanforderung saß in der Warteschlange zu lang. |
| 13812 | 0x000035F4 | ERROR_IPSEC_IKE_QUEUE_DROP_NO_MM | Aushandlungsanforderung saß in der Warteschlange zu lang. |
| 13813 | 0x000035F5 | ERROR_IPSEC_IKE_DROP_NO_RESPONSE | Keine Antwort von Peer. |
| 13814 | 0x000035F6 | ERROR_IPSEC_IKE_MM_DELAY_DROP | Verhandlung dauerte zu lange. |
| 13815 | 0x000035F7 | ERROR_IPSEC_IKE_QM_DELAY_DROP | Verhandlung dauerte zu lange. |
| 13816 | 0x000035F8 | ERROR_IPSEC_IKE_ERROR | Unbekannter Fehler aufgetreten. |
| 13817 | 0x000035F9 | ERROR_IPSEC_IKE_CRL_FAILED | Zertifikat Widerruf überprüfen konnte. |
| 13818 | 0x000035FA | ERROR_IPSEC_IKE_INVALID_KEY_USAGE | Ungültiges Zertifikat Schlüsselverwendung. |
| 13819 | 0x000035FB | ERROR_IPSEC_IKE_INVALID_CERT_TYPE | Ungültiges Zertifikat-Typ. |
| 13820 | 0x000035FC | ERROR_IPSEC_IKE_NO_PRIVATE_KEY | Kein Zertifikat zugeordneten privaten Schlüssel. |
| 13822 | 0x000035FE | ERROR_IPSEC_IKE_DH_FAIL | Fehler bei Diffie-Helman-Berechnung. |
| 13824 | 0x00003600 | ERROR_IPSEC_IKE_INVALID_HEADER | Ungültiger Header. |
| 13825 | 0x00003601 | ERROR_IPSEC_IKE_NO_POLICY | Keine Richtlinie konfiguriert. |
| 13826 | 0x00003602 | ERROR_IPSEC_IKE_INVALID_SIGNATURE | Fehler beim Überprüfen der Signatur. |

| Fehler | | | Beschreibung |
|---------|-------------|---------------------------------------|---|
| Dezimal | Hexadezimal | Name | |
| 13827 | 0x00003603 | ERROR_IPSEC_IKE_KERBEROS_ERROR | Fehler beim Authentifizieren mit Kerberos. |
| 13828 | 0x00003604 | ERROR_IPSEC_IKE_NO_PUBLIC_KEY | Zertifikat des Peers haben keinen öffentlichen Schlüssel. |
| 13829 | 0x00003605 | ERROR_IPSEC_IKE_PROCESS_ERR | Fehler Verarbeitung Fehler Nutzlast. |
| 13830 | 0x00003606 | ERROR_IPSEC_IKE_PROCESS_ERR_SA | Fehler Verarbeitung SA Nutzlast. |
| 13831 | 0x00003607 | ERROR_IPSEC_IKE_PROCESS_ERR_PROP | Fehler beim Verarbeiten Vorschlag Nutzlast. |
| 13832 | 0x00003608 | ERROR_IPSEC_IKE_PROCESS_ERR_TRANS | Fehler beim Verarbeiten Transform Nutzlast. |
| 13833 | 0x00003609 | ERROR_IPSEC_IKE_PROCESS_ERR_KE | Fehler Verarbeitung KE Nutzlast. |
| 13834 | 0x0000360A | ERROR_IPSEC_IKE_PROCESS_ERR_ID | Fehler-Verarbeitung-ID-Nutzlast. |
| 13835 | 0x0000360B | ERROR_IPSEC_IKE_PROCESS_ERR_CERT | Fehler beim Verarbeiten Cert Nutzlast. |
| 13836 | 0x0000360C | ERROR_IPSEC_IKE_PROCESS_ERR_CERT_REQ | Fehler beim Verarbeiten Zertifikatanforderung Nutzlast. |
| 13837 | 0x0000360D | ERROR_IPSEC_IKE_PROCESS_ERR_HASH | Fehler beim Verarbeiten Hash Nutzlast. |
| 13838 | 0x0000360E | ERROR_IPSEC_IKE_PROCESS_ERR_SIG | Fehler beim Verarbeiten Signatur Nutzlast. |
| 13839 | 0x0000360F | ERROR_IPSEC_IKE_PROCESS_ERR_NONCE | Fehler beim Verarbeiten Nonce Nutzlast. |
| 13840 | 0x00003610 | ERROR_IPSEC_IKE_PROCESS_ERR_NOTIFY | Fehler beim Verarbeiten Notify-Nutzlast. |
| 13841 | 0x00003611 | ERROR_IPSEC_IKE_PROCESS_ERR_DELETE | Fehler beim Verarbeiten der Nutzlast löschen. |
| 13842 | 0x00003612 | ERROR_IPSEC_IKE_PROCESS_ERR_VENDOR | Fehler beim Verarbeiten VendorId Nutzlast. |
| 13843 | 0x00003613 | ERROR_IPSEC_IKE_INVALID_PAYLOAD | Ungültige Nutzlast empfangen. |
| 13844 | 0x00003614 | ERROR_IPSEC_IKE_LOAD_SOFT_SA | Soft-Sicherheitszuordnung geladen. |
| 13845 | 0x00003615 | ERROR_IPSEC_IKE_SOFT_SA_TORN_DOWN | Soft-Sicherheitszuordnung abgerissen. |
| 13846 | 0x00003616 | ERROR_IPSEC_IKE_INVALID_COOKIE | Ungültige Cookie empfangen... |
| 13847 | 0x00003617 | ERROR_IPSEC_IKE_NO_PEER_CERT | Peer konnte gültiges Zertifikat senden. |
| 13848 | 0x00003618 | ERROR_IPSEC_IKE_PEER_CRL_FAILED | Zertifizierung Widerruf Kontrolle der Peerzertifikats ist fehlgeschlagen. |
| 13849 | 0x00003619 | ERROR_IPSEC_IKE_POLICY_CHANGE | Neue Richtlinie für ungültig erklärt SAs mit alten Politik gebildet. |
| 13850 | 0x0000361A | ERROR_IPSEC_IKE_NO_MM_POLICY | Es gibt keine verfügbaren Main Mode IKE-Richtlinie. |
| 13851 | 0x0000361B | ERROR_IPSEC_IKE_NOTCBPRIV | Fehler beim TCB-Berechtigung aktiviert. |
| 13852 | 0x0000361C | ERROR_IPSEC_IKE_SECLoadFAIL | Fehler beim Laden von Sicherheit.DLL. |
| 13853 | 0x0000361D | ERROR_IPSEC_IKE_FAILSSPINIT | Sicherheit Tabelle Versand Funktionsadresse von SSPI erhalten konnte. |
| 13854 | 0x0000361E | ERROR_IPSEC_IKE_FAILQUERYSSP | Fehler bei der Abfrage von Kerberos-Paket um max token-Größe zu erhalten. |
| 13855 | 0x0000361F | ERROR_IPSEC_IKE_SRVACQFAIL | Fehler beim Abrufen der Kerberos-Serveranmeldeinformationen für ISAKMP/
ERROR_IPSEC_IKE Dienst. Kerberos-Authentifizierung funktioniert nicht. Der wahrscheinlichste Grund dafür ist mangelnde Domänenmitgliedschaft. Dies ist normal, wenn Ihr Computer Mitglied einer Arbeitsgruppe ist. |
| 13856 | 0x00003620 | ERROR_IPSEC_IKE_SRVQUERYCRED | Fehler beim Bestimmen der SSPI-Prinzipalname für ISAKMP/ERROR_IPSEC_IKE Dienst (QueryCredentialsAttributes). |
| 13857 | 0x00003621 | ERROR_IPSEC_IKE_GETSPIFAIL | Konnte keine neue SPI für die eingehende SA vom Ipsec-Treiber zu erhalten. Die häufigste Ursache hierfür ist, dass der Treiber nicht den richtigen Filter. Überprüfen Sie Ihre Politik um die Filter zu überprüfen. |
| 13858 | 0x00003622 | ERROR_IPSEC_IKE_INVALID_FILTER | Da Filter ist ungültig. |
| 13859 | 0x00003623 | ERROR_IPSEC_IKE_OUT_OF_MEMORY | Speicherzuordnung fehlgeschlagen. |
| 13860 | 0x00003624 | ERROR_IPSEC_IKE_ADD_UPDATE_KEY_FAILED | Fehler beim Hinzufügen von Security Association zur IPsec-Treiber. Die häufigste Ursache hierfür ist, wenn die IKE-Aushandlung zu lange dauerte. Wenn das Problem weiterhin auftritt, reduzieren Sie die Auslastung auf dem fehlerhaften Computer. |
| 13861 | 0x00003625 | ERROR_IPSEC_IKE_INVALID_POLICY | Ungültige Richtlinien. |
| 13862 | 0x00003626 | ERROR_IPSEC_IKE_UNKNOWN_DOI | Ungültige DOI. |
| 13863 | 0x00003627 | ERROR_IPSEC_IKE_INVALID_SITUATION | Ungültige Lage. |
| 13864 | 0x00003628 | ERROR_IPSEC_IKE_DH_FAILURE | Diffie-Hellman-Fehler. |
| 13865 | 0x00003629 | ERROR_IPSEC_IKE_INVALID_GROUP | Ungültige Diffie-Hellman-Gruppe. |

| Fehler | | | Beschreibung |
|---------|-------------|---|--|
| Dezimal | Hexadezimal | Name | |
| 13866 | 0x0000362A | ERROR_IPSEC_IKE_ENCRYPT | Fehler verschlüsseln Nutzlast. |
| 13867 | 0x0000362B | ERROR_IPSEC_IKE_DECRYPT | Fehler entschlüsseln Nutzlast. |
| 13868 | 0x0000362C | ERROR_IPSEC_IKE_POLICY_MATCH | Fehler der Politik überein. |
| 13869 | 0x0000362D | ERROR_IPSEC_IKE_UNSUPPORTED_ID | Nicht unterstützte ID. |
| 13870 | 0x0000362E | ERROR_IPSEC_IKE_INVALID_HASH | Hashüberprüfung ist fehlgeschlagen. |
| 13871 | 0x0000362F | ERROR_IPSEC_IKE_INVALID_HASH_ALG | Ungültige Hash-Algorithmus. |
| 13872 | 0x00003630 | ERROR_IPSEC_IKE_INVALID_HASH_SIZE | Ungültige Hash-Größe. |
| 13873 | 0x00003631 | ERROR_IPSEC_IKE_INVALID_ENCRYPT_ALG | Ungültige Verschlüsselungsalgorithmus. |
| 13874 | 0x00003632 | ERROR_IPSEC_IKE_INVALID_AUTH_ALG | Ungültiger Authentifizierungsalgorithmus. |
| 13875 | 0x00003633 | ERROR_IPSEC_IKE_INVALID_SIG | Ungültiges Zertifikat-Signatur. |
| 13876 | 0x00003634 | ERROR_IPSEC_IKE_LOAD_FAILED | Laden fehlgeschlagen ist. |
| 13877 | 0x00003635 | ERROR_IPSEC_IKE_RPC_DELETE | Über RPC-Aufruf gelöscht. |
| 13878 | 0x00003636 | ERROR_IPSEC_IKE_BENIGN_REINIT | Vorübergehender Zustand erstellt, um Reinit auszuführen. Dies ist keine echte Fehler. |
| 13879 | 0x00003637 | ERROR_IPSEC_IKE_INVALID_RESPONDER_LIFETIME_NOTIFY | Den Lebenszeitwert erhielt der Responder Lifetime Benachrichtigen ist unter dem Mindestwert von Windows 2000 konfiguriert. Bitte korrigieren Sie die Richtlinie auf dem Peer-Computer. |
| 13880 | 0x00003638 | ERROR_IPSEC_IKE_QM_LIMIT_REAP | SA geerntet weil QM Grenze erreicht wurde. |
| 13881 | 0x00003639 | ERROR_IPSEC_IKE_INVALID_CERT_KEYLEN | Schlüssellänge im Zertifikat ist zu klein für die konfigurierten Sicherheitsanforderungen. |
| 13882 | 0x0000363A | ERROR_IPSEC_IKE_MM_LIMIT | Maximale Anzahl von etablierten MM SAs, peer überschritten. |
| 13883 | 0x0000363B | ERROR_IPSEC_IKE_NEGOTIATION_DISABLED | IKE erhielt eine Politik, die Aushandlung deaktiviert. |
| 13884 | 0x0000363C | ERROR_IPSEC_IKE_NEG_STATUS_END | |

6.8.2 TC_EVENT_CLASSES Konstanten

TC_EVENT_CLASSES ist eine globale TwinCAT System Variablenliste (GVL). Jede Konstante in dieser Liste identifiziert über den Variablennamen und den Variablenwert (GUID) eine auf einem TwinCAT System verfügbare Ereignisklasse bzw. Ereignisklassen-ID. Das TwinCAT System besitzt einige vordefinierte TwinCAT System Ereignisklassen-IDs. Weitere Ereignisklassen-IDs werden von den eingebundenen SPS-Bibliotheken und installierten TwinCAT Functions mitgebracht oder vom Anwender in der SPS-Applikation definiert. Die Variablenwerte (GUIDs) sind auf einem TwinCAT System eindeutig und werden vom TwinCAT System automatisch generiert und aktualisiert. Die Anzahl der verfügbaren Ereignisklassen-IDs kann dadurch von System zu System variieren.

Namensraum: TwinCAT TypeSystem [► 96].TC_EVENT_CLASSES

Typ: Globale Variablenliste (GVL)

Bibliothek: Nicht erforderlich. TwinCAT System Resource ab v3.1.4022.14

TwinCAT System Ereignisklassen-IDs

| Name | Typ |
|------------------------|------|
| TcSystemEventClass | GUID |
| TcGeneralAdsEventClass | GUID |
| TcRouterEventClass | GUID |
| TcRTimeEventClass | GUID |
| Win32EventClass | GUID |

TF6510 Ereignisklassen-IDs

| Name | Typ |
|---------------------------------------|------------|
| TcAcsiEventClass | GUID |
| TcIEC61850EventClass | GUID |
| TcScsmEventClass | GUID |
| | |
| TcMmsAbortEventClass | GUID |
| TcMmsAssociationEventClass | GUID |
| TcMmsClientEventClass | GUID |
| TcMmsIncomingConnectEventClass | GUID |
| TcMmsIncomingDisconnectEventClass | GUID |
| TcMmsOutgoingConnectEventClass | GUID |
| TcMmsOutgoingDisconnectEventClass | GUID |
| TcMmsReceiverEventClass | GUID |
| TcMmsSenderEventClass | GUID |
| TcMmsTransactionEventClass | GUID |
| | |
| TcUlosiAbortEventClass | GUID |
| TcUlosiAssociationEventClass | GUID |
| TcUlosiClientEventClass | GUID |
| TcUlosiIncomingConnectEventClass | GUID |
| TcUlosiIncomingDisconnectEventClass | GUID |
| TcUlosiOutgoingConnectEventClass | GUID |
| TcUlosiOutgoingDisconnectEventClass | GUID |
| TcUlosiReceiverEventClass | GUID |
| TcUlosiSenderEventClass | GUID |
| TcUlosiTransactionEventClass | GUID |
| | |
| TcRfc1006AbortEventClass | GUID |
| TcRfc1006AssociationEventClass | GUID |
| TcRfc1006ClientEventClass | GUID |
| TcRfc1006IncomingConnectEventClass | GUID |
| TcRfc1006IncomingDisconnectEventClass | GUID |
| TcRfc1006OutgoingConnectEventClass | GUID |
| TcRfc1006OutgoingDisconnectEventClass | GUID |
| TcRfc1006ReceiverEventClass | GUID |
| TcRfc1006SenderEventClass | GUID |
| TcRfc1006TransactionEventClass | GUID |
| | |
| TcTpktAbortEventClass | GUID |
| TcTpktAssociationEventClass | GUID |
| TcTpktPduEventClass | GUID |
| TcTpktReceiverEventClass | GUID |
| TcTpktSenderEventClass | GUID |
| TcTpktTransactionEventClass | GUID |
| | |
| TcSocketsAbortEventClass | GUID |
| TcSocketsAssociationEventClass | GUID |
| TcSocketsClientEventClass | GUID |
| TcSocketsIncomingConnectEventClass | GUID |

| Name | Typ |
|---------------------------------------|------|
| TcSocketsIncomingDisconnectEventClass | GUID |
| TcSocketsOutgoingConnectEventClass | GUID |
| TcSocketsOutgoingDisconnectEventClass | GUID |
| TcSocketsReceiverEventClass | GUID |
| TcSocketsSenderEventClass | GUID |
| TcSocketsTransactionEventClass | GUID |
| TcSocketsWin32SrvEventClass | GUID |

6.8.3 GVL_Acsi Konstanten

Namensraum: Tc3_Acsi [▶ 96].GVL_Acsi

Typ: Globale Variablenliste (GVL)

Bibliothek: Tc3_Acsi (Tc3_Acsi.compiled-library)

Funktionale Gruppen

| Konstante | Typ | Wert | Beschreibung |
|------------|-------|-------------|---|
| ACSI_FC_MX | DWORD | 16#00000001 | Messwerte (analogue value) |
| ACSI_FC_ST | DWORD | 16#00000002 | Statusinformation |
| ACSI_FC_CO | DWORD | 16#00000004 | Schaltersteuerung |
| ACSI_FC_CF | DWORD | 16#00000008 | Konfiguration |
| ACSI_FC_DC | DWORD | 16#00000010 | Beschreibung |
| ACSI_FC_SP | DWORD | 16#00000020 | Sollwerte |
| ACSI_FC_SG | DWORD | 16#00000040 | Setting-Group |
| ACSI_FC_RP | DWORD | 16#00000080 | Ungepufferte Reports |
| ACSI_FC_LG | DWORD | 16#00000100 | Protokollierung |
| ACSI_FC_BR | DWORD | 16#00000200 | Gepufferte Reports |
| ACSI_FC_GO | DWORD | 16#00000400 | Goose-Steuerung |
| ACSI_FC_GS | DWORD | 16#00000800 | Gsse-Steuerung |
| ACSI_FC_SV | DWORD | 16#00001000 | Ersatzwerte |
| ACSI_FC_SE | DWORD | 16#00002000 | Setting-Group-Editable |
| ACSI_FC_MS | DWORD | 16#00004000 | Multicast-Abtastwerte |
| ACSI_FC_SC | DWORD | 16#00008000 | SCL |
| ACSI_FC_US | DWORD | 16#00010000 | Unicast-Abtastwerte |
| ACSI_FC_EX | DWORD | 16#00020000 | Erweiterte Definition |
| ACSI_FC_XX | DWORD | 16#0003FFFF | Repräsentiert alle oben genannten Gruppen |

Auslösoptionen für Reports

| Konstante | Typ | Wert | Beschreibung |
|-----------------|-------|-------|--|
| ACSI_TrgOp_dchg | DWORD | 16#02 | Report bei Attributwertänderung |
| ACSI_TrgOp_qchg | DWORD | 16#04 | Report bei Änderung der Attributqualität |
| ACSI_TrgOp_dupd | DWORD | 16#08 | Report bei Attributwertaktualisierung |
| ACSI_TrgOp_intg | DWORD | 16#10 | Integrity-Report |
| ACSI_TrgOp_gi | DWORD | 16#20 | Report durch Generalabfrage |

Zugriffsberechtigung

| Konstante | Typ | Wert | Beschreibung |
|------------------|------------|-------------|------------------------------------|
| ACSI_AP_RO | BYTE | 1 | Lesezugriff |
| ACSI_AP_WO | BYTE | 2 | Schreibzugrif |
| ACSI_AP_RW | BYTE | 3 | Lese- und Schreibzugriff (Default) |

Weitere Konstanten

| Konstante | Typ | Wert | Beschreibung |
|------------------------|--------------------------|---|--|
| NULL_EntryID | T_OCTET8 | [8(0)] | Alle Oktette sind 0. |
| NULL_Owner | T_OCTET64 | [64(0)] | Alle Oktette sind 0. |
| NULL_TimeStamp | T_UtcTime | secondSinceEpoch:=DT#1970-01-01-00:00:00,
fractionOfSecond:=[0,0,0],
quality:=(LeapSecondsKnown:=0, ClockFailure:=0, ClockNotSynchronized:=0, Accuracy0:=0, Accuracy1:=0, Accuracy2:=0, Accuracy3:=0, Accuracy4:=0) | Zeitstempel mit dem Wert 0, alle „quality“-Bits sind 0 und „accuracy“ ist unbekannt. |
| NULL_EntryTime | T_BinaryTime | timeOfDay:=TOD#0:0:0, day:=0 | Erfassungszeit mit dem Wert 0. |
| NULL_Originator | ST_AcsiOriginator | orCat:=E_AcsiOrCategory.NotSupported, orIdent:=[64(0)] | Wert, „orCat“ ist 0 und alle Oktette im „orIdent“ sind 0. |
| NULL_ReasonCode | ST_AcsiReasonCode | reserved:=0, DataChange:=0, QualityChange:=0, DataUpdate:=0, Integrity:=0, GeneralInterrogation:=0, ApplicationTrigger:=0, unused:=0 | Alle Bits sind 0. |
| NULL_TriggerOps | ST_AcsiTriggerConditions | reserved:=0, DataChange:=0, QualityChange:=0, DataUpdate:=0, Integrity:=0, GeneralInterrogation:=0, reserved2:=0 | Alle Bits sind 0. |
| NULL_OptionalFields | ST_AcsiOptionalFields | reserved:=0, SequenceNumber:=0, ReportTimeStamp:=0, ReasonForInclusion:=0, DataSetName:=0, DataReference:=0, BufferOverflow:=0, EntryID:=0, ConfRevision:=0, Segmentation:=0 | Alle Bits sind 0. |
| NULL_LogOptionalFields | ST_AcsiLogOptionalFields | ReasonForInclusion:=0 | Alle Bits sind 0. |
| NULL_Quality | ST_AcsiQuality | Validity0:=0, Validity1:=0, Overflow:=0, OutOfRange:=0, BadReference:=0, Oscillatory:=0, Failure:=0, OldData:=0, Inconsistent:=0, Inaccurate:=0, Source:=0, Test:=0, OperatorBlocked:=0 | Alle Bits sind 0. |
| NULL_Unit | ST_AcsiUnit | SIUnit:=E_AcsiSIUnit.None, multiplier:=E_AcsiMultiplier.None | Wert, „SIUnit“ ist dimensionslos und „multiplier“ ist 0. |
| NULL_ValWithTrans | ST_AcsiValWithTrans | posVal:=0, transInd:=FALSE | Wert, „posVal“ ist 0 und „transInd“ ist FALSE. |
| NULL_AnalogueValue | ST_AcsiAnalogueValue | i:=0, f:=REAL#0.0 | Wert, „i“ ist 0 und „f“ ist 0. |
| NULL_UINT24 | T_UINT24 | [0,0,0] | Vorzeichenlose 24-Bit-Zahl mit dem Wert 0. |
| NULL_INT24 | T_INT24 | [0,0,0] | Vorzeichenbehaltete 24-Bit-Zahl mit dem Wert 0. |
| NULL_UINT128 | T_UINT128 | Hi:=0, Lo:=0 | Vorzeichenlose 128-Bit-Zahl mit dem Wert 0. |

| Konstante | Typ | Wert | Beschreibung |
|------------------|------------|--------------|--|
| NULL_INT128 | T_INT128 | Hi:=0, Lo:=0 | Vorzeichenbehaftete 128-Bit-Zahl mit dem Wert 0. |

7 Beispiele

In diesem Kapitel werden einige Beispiele zur Verwendung der IEC 61850 Implementierung in TwinCAT erläutert. Da sämtliche Beispiele auf einer allgemeinen Struktur aufbauen, wird diese vorerst erläutert und in den nachfolgenden Beispielen vorausgesetzt. Danach folgt für jedes Beispiel ein eigenes Kapitel, in welchem die speziellen Funktionalitäten des Projekts aufgeführt werden.

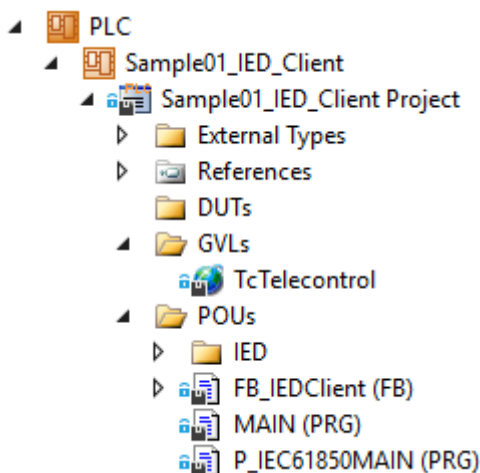
Der Beispielcode liegt der Dokumentation als zip-Dateien bei. Nach dem Entpacken der Beispiele ist es ratsam die Projekt-Ordnerstruktur beizubehalten. Jedem Beispiel liegt eine ICD-Datei bei. Wenn die ICD-Datei im TwinCAT Telecontrol Configurator aus dem Originalordner importiert und ein neues, modifiziertes TwinCAT-Projekt generiert wird, dann kann ein TwinCAT Projektvergleich mit dem Originalprojekt vorgenommen werden. TwinCAT Telecontrol Configurator wird während der Projekt-Codegenerierung eine entsprechende Meldung ausgeben und Projektvergleich vorschlagen. Voraussetzung dafür ist, dass in einem Unterverzeichnis (unterhalb der ICD-Datei) ein Projekt mit dem gleichen Namen gefunden wurde. Die Beispielprojekte verwenden die Default-Solution/Projekt-Bezeichnungen des TwinCAT Telecontrol Configurator. Um ein Projekt zu erzeugen, welches mit TwinCAT Project Compare verglichen werden kann muss bei der Codegenerierung bei den Server-Beispielen der Projekttyp: Server und bei Client-Beispielen der Projekttyp: Client ausgewählt werden.

Wenn die ICD-Datei an einen anderen Ort kopiert oder ein anderer Projekt- oder Solution-Name vergeben wurde, dann kann nach der Codegenerierung kein automatischer TwinCAT Projektvergleich vorgenommen werden. TwinCAT Telecontrol Configurator generiert an der neuen Stelle oder unter einem anderen Namen ein neues TwinCAT-Projekt. In diesem Fall können Sie TwinCAT Project Compare als Standalone-Tool starten (**Start > Beckhoff > TwinCAT Project Compare**), dort zu den beiden Projekten navigieren und ein Vergleich durchführen.

7.1 Allgemeine Client - Projektstruktur

Jedes der Client-Beispielprojekte folgt einer allgemeinen Struktur, welche in diesem Dokument erläutert werden soll. Zu dieser Struktur gehört die Verschachtelung der Funktionsbausteine im Solution-Explorer und der Aufbau der State-Machine im Client-Baustein, der die Verbindung zum Server herstellt und den Datenaustausch realisiert. Die Grundstruktur der TwinCAT IEC 61850 Beispielprojekte basiert auf der automatisch vom TwinCAT Telecontrol Configurator generierten SPS-Projektstruktur. Der TwinCAT Telecontrol Configurator verwendet bei der Generierung der Solution wiederum das „TwinCAT XAE Project (XML format)“ als Vorlage. Die in den Beispielen kommunizierten Datenmodelle spiegeln unterschiedliche IEC 61850 Server wider. Sie unterscheiden sich von Beispiel zu Beispiel und liegen dem Quellcode als icd-Dateien bei. Die ICD-Dateien können auch von einer Third-Party-Software zur Simulation eines Servers verwendet werden.

Zuerst eine kurze Betrachtung des Aufbaus eines TwinCAT IEC 61850 SPS-Client-Projekts:



Der generierte Solution-Name (wenn nicht anders angegeben) entspricht dem TwinCAT Telecontrol Configurator Projektname. Der automatisch generierte TwinCAT SPS-Projektname (wenn nicht anders angegeben) hat wiederum folgenden Aufbau: „[Projektname]_[IEDName]_Client“.

Standardmäßig besitzt jedes Beispielprojekt einen „DUTs“, „GVLs“ und „POUs“ Ordner. In dem „GVLs“-Ordner ist eine globale Variablenliste (GVL) mit dem Namen: „TcTelecontrol“ hinterlegt. In dieser globalen Variablenliste (siehe Quellcode unten) werden folgende Funktionsbausteine instanziiert und initialisiert:

- Eine Client-Bausteininstanz des Typs: FB_[IEDName]Client (Verbindungsmanagement und Datenaustausch).
- Eine IED-Datenmodell-Bausteininstanz des Typs: FB_IED_[IEDName].
- Optional (abhängig von der GOOSE-Subscriber-Konfiguration): Eine oder mehrere Bausteininstanzen des Typs: „FB_[IEDName]Gse“ für GOOSE-Kommunikation und GSE-Management.

Außerdem wird dort auch die während der Codegenerierung verwendete Codeversion und die verwendete Version des TwinCAT Telecontrol Configurator abgelegt.

Namensraum: TcTelecontrol

Typ: Globale Variablenliste (GVL)

```
VAR_GLOBAL
  ipCreator      : I_AcsiCodeCreatorClass := GVL_AcsiVars.Creator.SetCodeRev(codeRev:=2).SetGui
Ver(major:=1, minor:=0, build:=93, revision:=10);
  fb[IEDName]    : FB_IED_[IEDName];
  fb[IEDName]Client : FB_[IEDName]Client := (fbConnection:=(ipIED:=fb[IEDName], settings:=(sRemoteHost:='127.0.0.1')));
  fb[IEDName]Gse   : FB_[IEDName]Gse := (fbAdapter:=(ipIED:=fb[IEDName], settings:=(sMulticastAddr:='01-0C-CD-01-00-00', eDispatchMode:=E_GseDispatchMode.NonPromiscuous)));
END_VAR
```

Im „POUs“-Ordner befindet sich ein weiterer Ordner: „[IEDName]“, der die gesamte hierarchische Struktur des IED-Datenmodells als Funktionsbausteine beinhaltet. In diesem Ordner befindet sich auch der IED-Baustein, der in der globalen Variablenliste instanziiert wird und bereits erwähnt wurde. Auf der gleichen Ebene liegt der Client-Baustein: „FB_[IEDName]Client“, welcher den Verbindungsaufbau und den Datenaustausch mit einem IEC 61850 Server realisiert. Das TwinCAT SPS-Projekt enthält standardmäßig ein „MAIN“-Programm. Dieses wird zyklisch von einer TwinCAT Task aufgerufen und ruft wiederum das Programm „P_IEC61850MAIN“ auf. Das Programm „P_IEC61850MAIN“ kapselt den Aufruf des Client-Bausteins und des optionalen GSE-Bausteins, trennt die IEC 61850-Kommunikation von dem restlichen SPS-Maschinenprogramm und hilft z. B. bei der Implementierung weiterer Clients.

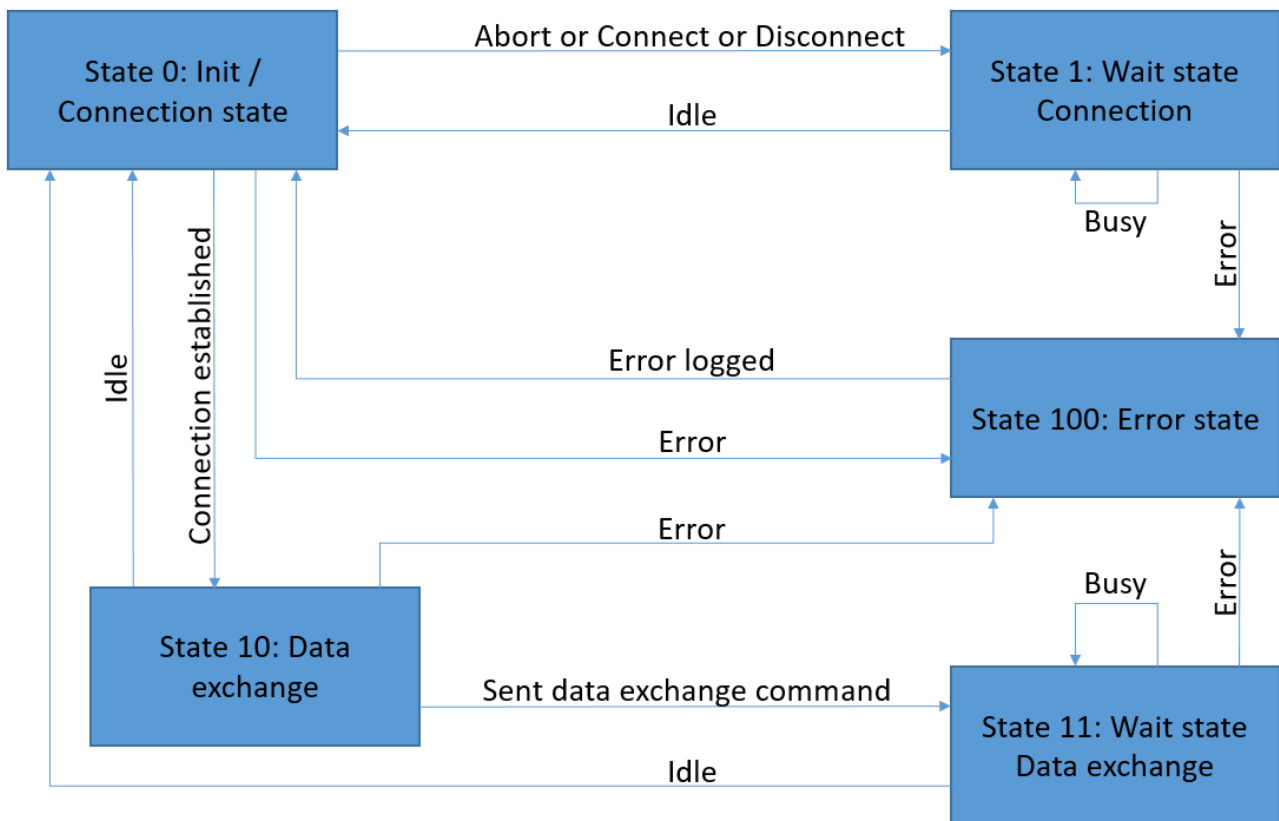
```
PROGRAM MAIN
VAR
END_VAR

P_IEC61850MAIN();

PROGRAM P_IEC61850MAIN
VAR
END_VAR

fb[IEDName]Client();
fb[IEDName]Gse();
```

In dem FB_[IEDName]Client-Baustein befindet sich eine State machine, von der die grundlegenden Zustände in jedem Client-Beispiel genutzt werden. Diese Zustände sind im folgenden Diagramm grafisch dargestellt:



State 0 (Init state): Die Statemachine befindet sich in diesem Zustand, sobald das SPS-Programm gestartet wurde. Hier (und im State 1) werden Befehle zur Verwaltung der Client-Server-Verbindung behandelt. Gesteuert wird dies hauptsächlich über vier boolesche Variablen. Diese Variablen, wenn sie gesetzt sind, aktivieren die entsprechenden Befehle (in diesem Fall sind es einmalige Methodenaufrufe am Client-Baustein).

- **_bAbort:** Ruft die Methode „AbortReq“ auf, welche den Befehl zum Abbruch der Client-Verbindung zum Server aktiviert.
- **_bConnect:** Ruft die Methode „AssociateReq“ auf, welche den Befehl zum Aufbau einer neuen Client-Verbindung zum Server aktiviert.
- **_bDisconnect:** Ruft die Methode „ReleaseReq“ auf, welche den Befehl zum geregelten Abbau einer bestehenden Client-Verbindung zum Server aktiviert.
- **_bReconnect:** Ruft ebenfalls die Methode „AssociateReq“ auf, falls die Client-Verbindung zum Server abgebrochen/getrennt wurde, aber automatisch wiederhergestellt werden soll.

Die zuvor aufgezählten Methoden, welche in diesem State einmalig aufgerufen werden, benötigen für die Abarbeitung länger als einen SPS-Zyklus. Aus diesem Grund wechselt die Statemachine in einen Warte-Zustand (State 1), in dem auf die Beendigung des aktivierten Befehls gewartet wird.

Falls die Client-Verbindung zum Server zuvor bereits hergestellt wurde, wird die Statemachine in den Zustand zur Datenübertragung wechseln (State 10).

State 1 (Wait State): In diesem Zustand verweilt der Client bis die Befehlsabarbeitung zur Verwaltung der Client-Server-Verbindung nicht mehr beschäftigt (busy) ist. Solange die Verbindung hergestellt, abgebaut oder abgebrochen wird, befindet sich die Statemachine im State 1. Sobald der Befehl erfolgreich abgearbeitet wurde, wird die Statemachine zurück in den State 0 (Init State) versetzt.

State 10 (Data exchange): Wenn die Statemachine in diesem Zustand ist, dann ist die Client-Verbindung zum Server bereits erfolgreich hergestellt worden. Der Client ist bereit für die Datenübertragung zum Server. Während der Datenübertragung werden Befehle zum Versenden oder Empfangen der Daten aktiviert.

In diesem Zustand unterscheiden sich die Beispielprojekte untereinander. Abhängig von der gewünschten Funktionalität oder Logik in der Applikation werden hier unterschiedliche Methoden oder Hilfsbausteine aufgerufen. Außerdem kann der Client-Baustein um weitere States erweitert werden.

Sämtliche Methodenaufrufe, die Befehle zur Datenübertragung aktivieren, benötigen für eine erfolgreiche Abarbeitung mehrere SPS-Zyklen, daher muss die Statemachine, nach dem Aufruf einer solchen Methode, in den State 11 (Wait state) versetzt werden.

Bei einer aktiven Verbindung und keiner Datenübertragung wechselt die Statemachine zwischen dem State 0 und 10. Die Statemachine wird wieder in den State 0 versetzt, um auf Änderungen im Client-Server-Verbindungsstatus zu reagieren und sie im State 0 zu behandeln.

State 11 (Wait state): Dieser Zustand ist ein weiterer Warte-Zustand. Sobald ein Befehl zur Datenübertragung (aktiviert im State 10) abgearbeitet wurde, wird die Statemachine in den State 11 versetzt und wartet bis die Befehlsabarbeitung nicht mehr beschäftigt (busy) ist. Danach wechselt die Statemachine in den State 0.

State 100 (Error state): Sobald ein Fehler bei der Aktivierung oder Abarbeitung eines Befehls auftritt, wird die Statemachine in den State 100 versetzt. Hier wird der Fehler geloggt und die Statemachine zurück in den State 0 versetzt.

```

FUNCTION_BLOCK FB_[IEDName]Client
VAR_INPUT
    fbConnection          : FB_iec61850ClientClass;
END_VAR
VAR
    _bAbort              : BOOL;
    _bDisconnect         : BOOL;
    _bConnect            : BOOL;
    _bReconnect          : BOOL := TRUE;
    _bReadAllData        : BOOL := TRUE;
    state                : BYTE;
    eState               : E_AsyncEnvironmentState;
    bBusy                : BOOL;
    bSuccess             : BOOL;
    ipResult             : I_AsyncServiceResultClass;
    sLastErrorResult     : T_MaxString;
    fbAbortReason        : FB_ServiceErrorClass := (stError:=SUCCESS_EVENT);
    sLastAbortReason     : T_MaxString;
    nInvokeID            : UDINT;
    eServiceError        : E_AcsiServiceError;
    nServiceError        : UDINT;
    nCmdError            : UDINT;
    sObjReference        : T_AcsiObjectReference;
    sCtrlReference       : T_AcsiObjectReference;

    bGetServerDirectory : BOOL := TRUE;
    bGetLogicalDeviceDirectory : BOOL := TRUE;
    bGetLogicalNodeDirectory : BOOL := TRUE;
    bGetAllServerValues  : BOOL := TRUE;
    bGetAllServerValues  : BOOL := TRUE;
    bGetAllDataValues_LLNO_ST : BOOL := TRUE;
    bGetDataValues_LLNO_ST_Beh : BOOL := TRUE;
END_VAR

fbConnection.Execute();
eState:= fbConnection.eState;

CASE state OF
    0:
        IF _bAbort THEN
            _bAbort:= FALSE;
            bSuccess:= fbConnection.AbortReq(ipReason:=fbAbortReason, ipSink:=0, ipResult=>ipResult)
;
            state:= SEL(bSuccess, 100, 1);
        ELSIF eState = E_AsyncEnvironmentState.Idle AND (_bConnect OR _bReconnect) THEN
            _bConnect:= FALSE;
            bGetAllServerValues:= SEL(_bReadAllData, bGetAllServerValues, TRUE);
            bSuccess:= fbConnection.AssociateReq(ipSink:=0, ipResult=>ipResult);
            state:= SEL(bSuccess, 100, 1);
        ELSIF eState = E_AsyncEnvironmentState.Established AND _bDisconnect THEN
            _bDisconnect:= FALSE;
            bSuccess:= fbConnection.ReleaseReq(ipSink:=0, ipResult=>ipResult);
            state:= SEL(bSuccess, 100, 1);
        ELSIF eState = E_AsyncEnvironmentState.Established THEN
            state:= 10;
        END_IF
        _bConnect:= FALSE;
        _bDisconnect:= FALSE;
    1:
        IF ipResult <> 0 THEN

```

```

        ipResult.Execute();
        IF NOT (bBusy:=ipResult.IsBusy()) THEN
            state:= SEL(ipResult.IsCompleted(), 100, 0);
        END_IF
    END_IF
10:
    IF bGetServerDirectory THEN
        bGetServerDirectory:= FALSE;
        bSuccess:= fbConnection.GetServerDirectoryReq(ipServer:=fb[IEDName], eClass:=E_AcsiServerDirectoryClass.LogicalDevice, hUser:=0, ipSink:=0, nInvokeID=>nInvokeID, ipResult=>ipResult);
        state:= SEL(bSuccess, 100, 11);
    ELSIF bGetLogicalDeviceDirectory THEN
        bGetLogicalDeviceDirectory:= FALSE;
        bSuccess:= fbConnection.GetLogicalDeviceDirectoryReq(ipLogicalDevice:=fb[IEDName].IEDLD1, hUser:=0, ipSink:=0, nInvokeID=>nInvokeID, ipResult=>ipResult);
        state:= SEL(bSuccess, 100, 11);
    ELSIF bGetLogicalNodeDirectory THEN
        bGetLogicalNodeDirectory:= FALSE;
        bSuccess:= fbConnection.GetLogicalNodeDirectoryReq(ipLogicalNode:=fb[IEDName].IEDLD1.LLN0, eClass:=E_AcsiLogicalNodeClass.DataSet, hUser:=0, ipSink:=0, nInvokeID=>nInvokeID, ipResult=>ipResult);
        state:= SEL(bSuccess, 100, 11);
    ELSIF bGetAllServerValues THEN
        bGetAllServerValues:= FALSE;
        bSuccess:= fbConnection.GetAllServerValuesReq(ipServer:=fb[IEDName], hUser:=0, ipSink:=0, nInvokeID=>nInvokeID, ipResult=>ipResult);
        state:= SEL(bSuccess, 100, 11);
    ELSIF bGetAllDataValues_LLNO_ST THEN
        bGetAllDataValues_LLNO_ST:= FALSE;
        bSuccess:= fbConnection.GetAllDataValuesReq(ipLogicalNode:=fb[IEDName].IEDLD1.LLN0, eFc:=E_AcsiFc.ST_, hUser:=0, ipSink:=0, nInvokeID=>nInvokeID, ipResult=>ipResult);
        state:= SEL(bSuccess, 100, 11);
    ELSIF bGetDataValues_LLNO_ST_Beh THEN
        bGetDataValues_LLNO_ST_Beh:= FALSE;
        bSuccess:= fbConnection.GetDataValuesReq(ipData:=fb[IEDName].IEDLD1.LLN0.Beh, eFc:=E_AcsiFc.ST_, hUser:=0, ipSink:=0, nInvokeID=>nInvokeID, ipResult=>ipResult);
        state:= SEL(bSuccess, 100, 11);
    ELSE
        state:= 0;
    END_IF
11:
    IF ipResult <> 0 THEN
        ipResult.Execute();
        IF NOT (bBusy:=ipResult.IsBusy()) THEN
            state:= SEL(ipResult.IsCompleted(), 100, 0);
        END_IF
    END_IF
100:
    state:= 0;
    IF ipResult <> 0 THEN
        nCmdError:= nCmdError + 1;
        sLastErrorResult:= ipResult.Dump();
    END_IF
END_CASE

```

GOOSE-Subscriber (optional)

TwinCAT Telecontrol Configurator kann in einem Client-Projekt, während der SPS-Code-Generierung den SPS-Code für einen GOOSE-Subscriber ebenfalls generieren (siehe Codebeispiel unten). Dies setzt voraus, dass der Anwender vorher die GOOSE-Komponenten wie z.B. GoCBs (goose control blocks) im TwinCAT Telecontrol Configurator erstellt oder aus einer SCL-Datei (z. B. ICD-Datei) importiert hat.

Standardmäßig wird bei der Codegenerierung ein Baustein mit dem Namen: „FB_[IEDName]Gse“ instanziiert und der globalen Variablenliste „TcTelecontrol“ hinzugefügt. Dieser Baustein stellt die Verbindung zwischen einem Netzwerkadapter des TwinCAT Steuerungsrechners, dem IED-Datenmodell und der GOOSE-Konfiguration in den GoCBs her. Die GoCBs werden in dem IED-Datenmodell (normalerweise im LLN0) instanziiert. Jeder GoCB besitzt ein Baustein-Unterelement mit dem Namen: „Subscriber“. Über die „Subscriber“-Methodenaufrufe kann die Subscription aus dem SPS-Code aktiviert („Enable“) bzw. deaktiviert werden („Disable“). Defaultmäßig wird die Subscription beim SPS-Programmstart für alle GoCBs aktiviert. Dies wird durch die mit „TRUE“ initialisierte „bSubscriber“-Variable gesteuert. Über eine steigende Flanke an der „bUnsubscribe“-Variablen kann die Subscription für alle GoCBs deaktiviert werden. Die über diese Methoden abgesetzten Subscriber-Befehle werden sofort ausgeführt, ohne Wartezyklen oder weitere States die nötig sind, um die Befehlsabarbeitung abzuschließen.

Der hier beschriebene „Subscriber“ liest die Konfiguration und aktualisiert zwar den Status des GoCBs (Attribute „GoEna“ wird z.B. auf „TRUE“ oder „FALSE“ gesetzt), er nutzt aber nicht die Client-Server-Dienste wie „SetGoCBValues“ oder „GetGoCBValues“, um den „Publisher“ auf der Serverseite zu aktivieren bzw. zu

deaktivieren. D.h. der generierte Code implementiert bereits einen Subscriber der z. B. im ersten SPS-Zyklus oder aus dem SPS-Code zum beliebigen Zeitpunkt aktiviert bzw. deaktiviert werden kann. Die benötigten GoCB-Konfigurationseinstellungen (GoCB-Attributwerte) können über Initialisierungswerte vorgenommen werden. Die Konfiguration der GoCBs kann aber bereits in dem TwinCAT Telecontrol Configurator geschehen. Die Initialisierungswerte werden während der Codegenerierung automatisch generiert und zugewiesen. Wenn der Subscriber aktiviert wurde und die Konfiguration des GoCBs und des Netzwerkadapters zu dem empfangenden GOOSE-Frame eine Übereinstimmung ergibt, werden die GOOSE-Daten ins TwinCAT IED-Datenmodell umkopiert (gemappt). Die „Execute“-Methode muss in der übrigen Zeit zyklisch aufgerufen werden. Sie ist unter anderem für die Aktualisierung der Statusinformation im GoCB verantwortlich.

```
FUNCTION_BLOCK FB_[IEDName]Gse IMPLEMENTS I_GseLinkStatusEventSink
VAR_INPUT
    fbAdapter      : FB_GseAdapterClass := (ipLinkStatus:=THIS^);
END_VAR
VAR
    eLinkStatus   : E_GseLinkStatus;
    bSuccess       : BOOL;
    ipError        : I_ServiceErrorClass;
    bSubscribe     : BOOL := TRUE;
    bUnsubscribe  : BOOL;
END_VAR

bSuccess:= fbAdapter.Execute(ipError=>ipError);
IF bSubscribe THEN
    bSubscribe:= FALSE;
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb01.Subscriber.Enable(ipAdapter:=fbAdapter, ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb02.Subscriber.Enable(ipAdapter:=fbAdapter, ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb03.Subscriber.Enable(ipAdapter:=fbAdapter, ipError=>ipError);
ELSIF bUnsubscribe THEN
    bUnsubscribe:= FALSE;
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb01.Subscriber.Disable(ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb02.Subscriber.Disable(ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb03.Subscriber.Disable(ipError=>ipError);
ELSE
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb01.Subscriber.Execute(ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb02.Subscriber.Execute(ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb03.Subscriber.Execute(ipError=>ipError);
END_IF
```

Im Projektbaum unter dem Zweig I/O-Device finden Sie eine Netzwerkadapterinstanz mit dem Namen „GSE (RT-Ethernet Adapter)“. Diese Adapterinstanz muss entsprechend konfiguriert werden, d.h. die I/O-Konfiguration muss an die vorhandene Hardware und auf die Zielplattform, auf der das Projekt laufen soll, angepasst werden.

Eine erneute I/O-Konfiguration ist auch dann nötig, wenn Sie die Zielplattform wechseln. Diese Konfiguration muss manuell in TwinCAT XAE vorgenommen werden. Zusätzlich zu der I/O-Konfiguration des Netzwerkadapters muss ein Link zwischen dem Netzwerkadapter und den SPS-Bausteinen für die Goose-Kommunikation hergestellt werden. Mit dem Link können die vom Netzwerkadapter empfangenen Daten an die Instanz des Funktionsbausteins: „FB_[IEDName]Gse“ weitergeleitet werden. In umgekehrter Richtung kann die Instanz des Funktionsbausteins „FB_[IEDName]Gse“ die zu sendenden Daten an den Netzwerkadapter weiterleiten.

Hier finden Sie weitere Informationen: [RT – Ethernet Adapter Konfiguration \[► 37\]](#).

7.2 Client - Basis Beispielprojekt

Dieses Beispiel zeigt die grundlegende TwinCAT Implementierung der Kommunikation eines TwinCAT IEC 61850 Clients und die Verwendung einiger Methoden, um Daten von einem Server zu lesen. Die GOOSE-Komponenten und Reporting sind in diesem Beispiel nicht enthalten, um es einfach halten zu können. Diese und andere Funktionalitäten werden in separaten Beispielen behandelt. Der hier implementierte Client versucht nach dem SPS-Programmstart eine Verbindung zum Server mit der Host-Adresse: „127.0.0.1“ und Portnummer: 102 aufzubauen. Diese Werte können im SPS-Beispielcode angepasst werden. Die SPS-Entwicklungsumgebung wird beim Übersetzen eine Warnung mit einem Hinweis dazu ausgeben. Ein Doppelklick auf diese Meldung leitet Sie zu der entsprechenden Stelle im SPS-Code.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1031/TF6510_TC3_IEC61850/Resources/5494779659.zip

Das hier beschriebene Beispiel nutzt die State Machine, welche in dem Kapitel „Allgemeine Client - Projektstruktur [▶ 591]“ beschrieben ist. Die States: 0, 1, 11 und 100 sind identisch zu der dort beschriebenen State Machine. Andere States wurden für das Beispiel modifiziert oder auch neue States hinzugefügt.

Im State 10 sind mehrere IF-Anweisungen aufgeführt, welche Befehle (Methodenaufrufe) des Client-Bausteins aktivieren. Durch eine steigende Flanke an einer der booleschen Variablen in der IF-Anweisung wird der Befehl aktiviert und die entsprechende Client-Methode aufgerufen. Zur Testzwecken kann der Benutzer in der Online-Ansicht die Werte der booleschen Variablen auf „TRUE“ setzen und die Befehlsabarbeitung auf diese Weise auslösen.

Nach der erfolgreichen Abarbeitung eines Befehls wird die State Machine in den State 0 versetzt. Falls mehrere der booleschen Variablen auf „TRUE“ gesetzt wurden, wird die oberste IF-Anweisung mit dem jeweiligen Befehl zuerst bearbeitet. Je häufiger und komplexer die Befehle, die an den Server gesendet werden, desto länger benötigt der Client, um diese abzuarbeiten.

In der folgenden Tabelle sind einige der booleschen Variablen aufgelistet. Zudem sind die Methodennamen und eine Beschreibung der Funktionalität enthalten.

Befehle im Sample01

| Variablenname | Methodenname | Beschreibung |
|----------------------------|------------------------------|--|
| bGetServerDirectory | GetServerDirectoryReq | Gibt eine Liste aller logischen Geräte (Logical Devices) eines IEDs (Intelligent Electronic Device) zurück. |
| bGetLogicalDeviceDirectory | GetLogicalDeviceDirectoryReq | Gibt eine Liste aller logischen Knoten (Logical Nodes) innerhalb eines logischen Gerätes (Logical Device) zurück. |
| bGetLogicalNodeDirectory | GetLogicalNodeDirectoryReq | Gibt eine Liste von allen Instanzen einer bestimmten logischen Knoten Klasse (Logical Node Class) wieder. In unserem Beispiel ist es die Klasse: „E_AcsiLogicalNodeClass.DataSet“. |
| bGetAllServerValues | GetAllServerValuesReq | Liest alle Datenwerte des angegebenen IEDs. |
| bGetAllDataValues_LLNO_ST | GetAllDataValuesReq | Liest alle Datenwerte eines angegebenen logischen Knoten (Logical Node) und der angegebenen funktionalen Gruppe (Functional Constraint). |
| bGetDataValues_LLNO_ST_Beh | GetDataValuesReq | Liest alle Datenwerte eines angegebenen Datenobjekts (Data Object) und der angegebenen funktionalen Gruppe (Functional Constraint). |

Die Benennung der booleschen Variablen ist beispielhaft und kann beliebig angepasst werden. Sie soll nur verdeutlichen, welche Werte durch das Anschalten der Variablen gelesen werden können.

7.3 Client - Read Data (GetAllDataValues, GetDataValues)

Dieses Beispiel zeigt die Verwendung der „GetAllDataValuesReq“- und „GetDataValuesReq“-Methoden des Client-Funktionsbausteins.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1031/TF6510_TC3_IEC61850/Resources/5494782347.zip

Das hier beschriebene Beispiel nutzt die State Machine, welche in dem Kapitel „Allgemeine Client - Projektstruktur [▶ 591]“ beschrieben ist. Die States: 0, 1, 11 und 100 sind identisch zu der dort beschriebenen State Machine. Andere States wurden für das Beispiel modifiziert oder auch neue States hinzugefügt.

Im State 10 sind mehrere IF-Anweisungen aufgeführt, welche Befehle (Methodenaufrufe) des Client-Bausteins aktivieren. Durch eine steigende Flanke an einer der booleschen Variablen in der IF-Anweisung wird der Befehl aktiviert und die entsprechende Client-Methode aufgerufen. Zu Testzwecken kann der Benutzer in der Online-Ansicht die Werte der booleschen Variablen auf „TRUE“ setzen und die Befehlsabarbeitung auf diese Weise auslösen.

Nach der erfolgreichen Abarbeitung eines Befehls wird die State Machine in den State 0 versetzt. Falls mehrere der booleschen Variablen auf „TRUE“ gesetzt wurden, wird die oberste IF-Anweisung mit dem jeweiligen Befehl zuerst bearbeitet. Je häufiger und komplexer die Befehle, die an den Server gesendet werden, desto länger benötigt der Client diese abzuarbeiten.

In der folgenden Tabelle sind einige der booleschen Variablen aufgelistet. Zudem sind die Methodennamen und eine Beschreibung der Funktionalität enthalten.

Befehle im Sample02

| Variablenname | Methodenname | Beschreibung |
|---|---------------------|---|
| bGetAllDataValues_LLNO
ähnliche Variablen (Befehle):
bGetAllDataValues_LPHD1
bGetAllDataValues_MMXU1
bGetAllDataValues_XCBR1 | GetAllDataValuesReq | Sobald diese boolesche Variable auf „TRUE“ gesetzt wird, werden alle Datenwerte sämtlicher Unter-elemente des logischen Knoten: „LLNO“ und jeder funktionalen Gruppe gelesen. Um dies zu ermöglichen, verwendet der Methodenaufruf als Parameter einen Schnittstellenzeiger auf den zu lesenden logischen Knoten: „LLNO“ und die Angabe der funktionalen Gruppe: „E_AcsiFc.XX.“ |
| bGetAllDataValues_LLNO_ST
ähnliche Variablen (Befehle):
bGetAllDataValues_LLNO_CF
bGetAllDataValues_LLNO_DC
bGetAllDataValues_LLNO_EX
bGetAllDataValues_LPHD1_ST
bGetAllDataValues_LPHD1_DC
bGetAllDataValues_MMXU1_MX
bGetAllDataValues_MMXU1_ST
bGetAllDataValues_XCBR1_ST
bGetAllDataValues_XCBR1_CO
bGetAllDataValues_XCBR1_CF | GetAllDataValuesReq | Sobald diese boolesche Variable auf „TRUE“ gesetzt wird, werden alle Datenwerte sämtlicher Unter-elemente des logischen Knoten: „LLNO“ und der funktionalen Gruppe: „ST“ gelesen. Um dies zu ermöglichen, verwendet der Methodenaufruf als Parameter einen Schnittstellenzeiger auf den zu lesenden logischen Knoten: „LLNO“ und die Angabe der funktionalen Gruppe: „E_AcsiFc.ST.“. |
| bGetDataValues_LLNO_ST_Beh
ähnliche Variablen (Befehle):
bGetDataValues_LLNO_ST_Health
bGetDataValues_LLNO_ST_Mod
bGetDataValues_LLNO_CF_Mod
bGetDataValues_LLNO_DC_NamPlt
bGetDataValues_LLNO_EX_NamPlt
bGetDataValues_LPHD1_ST_PhyHealth
bGetDataValues_LPHD1_ST_Proxy
bGetDataValues_LPHD1_DC_PhyNam
bGetDataValues_MMXU1_MX_PhV
bGetDataValues_MMXU1_ST_Beh
bGetDataValues_XCBR1_ST_Beh | GetDataValuesReq | Sobald diese boolesche Variable auf „TRUE“ gesetzt wird, werden alle Datenwerte sämtlicher Unter-elemente des Datenobjektes: „Beh“ des logischen Knoten: „LLNO“ und der funktionalen Gruppe: „ST“ gelesen. Um dies zu ermöglichen, verwendet der Methodenaufruf als Parameter einen Schnittstellenzeiger auf das zu lesende Datenobjekt: „LLNO.Beh“ und die Angabe der funktionalen Gruppe: „E_AcsiFc.ST.“. |
| bGetDataValues_LLNO_ST_Beh_stVal
ähnliche Variablen (Befehle):
bGetDataValues_LLNO_ST_Beh_q
bGetDataValues_LLNO_ST_Beh_t
bGetDataValues_LLNO_ST_Health_q
bGetDataValues_LLNO_ST_Health_t
bGetDataValues_LLNO_ST_Mod_q
bGetDataValues_LLNO_ST_Mod_t
bGetDataValues_LLNO_CF_Mod_ctlModel
bGetDataValues_LLNO_DC_NamPlt_vendor
bGetDataValues_LLNO_DC_NamPlt_swRev | GetDataValuesReq | Sobald diese boolesche Variable auf „TRUE“ gesetzt wird, wird der Wert des Datenattributs: „stVal“ des Datenobjektes: „Beh“, des logischen Knoten: „LLNO“ und der funktionalen Gruppe: „ST“ gelesen. Um dies zu ermöglichen, verwendet der Methodenaufruf als Parameter einen Schnittstellenzeiger auf das zu lesende Datenattribut: „LLNO.Beh.stVal“ und die Angabe der funktionalen Gruppe: „E_AcsiFc.ST.“. |

Die Benennung der booleschen Variablen ist beispielhaft und kann beliebig angepasst werden. Sie soll nur verdeutlichen, welche Werte durch das Anschalten der Variablen gelesen werden können.

7.4 Client - Write Data (SetAllDataValues, SetDataValues)

Dieses Beispiel zeigt die Verwendung der „SetAllDataValuesReq“- und „SetDataValuesReq“-Methoden des Client-Funktionsbausteins.

Download TwinCAT XAE Projekt (*.zip): https://infosys.beckhoff.com/content/1031/TF6510_TC3_IEC61850/Resources/8922555275.zip

Das hier beschriebene Beispiel nutzt die Statemachine, welche in dem Kapitel „Allgemeine Client - Projektstruktur [► 591]“ beschrieben ist. Die States: 0,1,11 und 100 sind identisch zu der dort beschriebenen Statemachine. Andere States wurden für das Beispiel modifiziert oder auch neue States hinzugefügt.

Beispielprojekt

Im Funktionsbaustein FB_IEDClient und State 10 sind mehrere IF-Anweisungen aufgeführt, welche Befehle (Methodenaufrufe) des Client-Bausteins aktivieren. Jeder Befehl in der IF-Anweisung wird durch eine steigende Flanke an einer der booleschen Variablen aktiviert. Die booleschen Variablen sind im Deklarationsteil des Funktionsbausteins „FB_IEDClient“ definiert. Zu Testzwecken kann der Benutzer in der Online-Ansicht die Werte der booleschen Variablen auf „TRUE“ setzen und die Befehlsabarbeitung auf diese Weise aktivieren. Danach wird zum State 11 gewechselt, wo in weiteren SPS-Zyklen die Methode „ipResult.Execute()“ so lange aufgerufen wird, bis die Befehlsabarbeitung abgeschlossen wurde. Dies ist dann der Fall, wenn „ipResult.IsBusy()“ den Wert „FALSE“ zurück liefert.

Nach der erfolgreichen Abarbeitung eines Befehls wird die Statemachine in den State 0 versetzt. Falls mehrere der booleschen Variablen auf „TRUE“ gesetzt sind, dann wird die oberste IF-Anweisung mit dem jeweiligen Befehl zuerst abgearbeitet. In der folgenden Tabelle sind einige der booleschen Variablen aufgelistet. Zudem sind die Methodennamen und eine Beschreibung der Funktionalität enthalten.

| Variablenname | Methodenname | Beschreibung |
|--|---------------------|--|
| bSetAllDataValues_LPHD1_DC

ähnliche Variablen (Befehle):
bSetAllDataValues_LLNO_DC
bSetAllDataValues_MMXU1_DC
bSetAllDataValues_XCBR1_DC
bSetAllDataValues_CSWI1_DC
bSetAllDataValues_LEDGGIO1_DC
bSetAllDataValues_LEDGGIO2_DC
bSetAllDataValues_LEDGGIO3_DC
bSetAllDataValues_LEDGGIO4_DC
bSetAllDataValues_LEDGGIO5_DC
bSetAllDataValues_LEDGGIO6_DC
bSetAllDataValues_LEDGGIO7_DC
bSetAllDataValues_LEDGGIO8_DC | SetAllDataValuesReq | Sobald diese boolesche Variable auf „TRUE“ gesetzt wird, werden alle Datenwerte sämtlicher Unterelemente des logischen Knoten: „LPHD1“ und der funktionalen Gruppe „DC“ geschrieben. Um dies zu ermöglichen, verwendet der Methodenaufruf als Parameter einen Schnittstellenzeiger auf den zu schreibenden logischen Knoten: „LPHD1“ und die Angabe der funktionalen Gruppe: „E_AcsiFc.DC“. |
| bSetDataValues_MMXU1_CF_TotW

ähnliche Variablen (Befehle):
bSetDataValues_LEDGGIO1_CF_AnOut1_sboClass
bSetDataValues_LEDGGIO1_CF_AnOut1_ctlM odel
bSetDataValues_LEDGGIO1_CF_AnOut1_units
bSetDataValues_LEDGGIO1_CF_AnOut1_units_SUnit
bSetDataValues_LEDGGIO1_CF_AnOut1_units_multiplier | SetDataValuesReq | Sobald diese boolesche Variable auf „TRUE“ gesetzt wird, werden alle Datenwerte sämtlicher Unterelemente des Datenobjektes: „TotW“ des logischen Knoten „MMXU1“ und der funktionalen Gruppe: „CF“ geschrieben. Um dies zu ermöglichen, verwendet der Methodenaufruf als Parameter einen Schnittstellenzeiger auf das zu schreibende Datenobjekt: „MMXU1.TotW“ und die Angabe der funktionalen Gruppe: „E_AcsiFc.CF“. |

Die Benennung der booleschen Variablen ist beispielhaft und kann beliebig angepasst werden. Sie soll nur verdeutlichen, welche Werte durch das Anschalten der Variablen geschrieben werden können.

7.5 Client - Report Control Blocks (Unbuffered, Buffered)

Dieses Beispiel zeigt die Verwendung der Berichtssteuerblockinstanzen für gepufferte und ungepufferte Reports (Unbuffered/Buffered Report Control Blocks: URCB, BRCB). Dazu gehört das Aktivieren und Deaktivieren von Reports oder das Aktivieren der Generalabfrage (GI).

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1031/TF6510_TC3_IEC61850/Resources/5494785035.zip

Das hier beschriebene Beispiel nutzt die State Machine, welche in dem Kapitel „Allgemeine Client - Projektstruktur [► 591]“ beschrieben ist. Die States: 0, 1, 11 und 100 sind identisch zu der dort beschriebenen State Machine. Andere States wurden für das Beispiel modifiziert oder auch neue States hinzugefügt.

Die Report Control Blocks (RCBs) konfigurieren und steuern das Versenden der gepufferten oder ungepufferten Reportmeldungen. Die Konfiguration und Steuerung der Reports erfolgt über Schreib-/Lesezugriffe auf die entsprechenden Attributwerte eines Report Control Blocks. Einige Attributwerte konfigurieren z. B. den Inhalt der im Report enthaltenen Information, andere Attributwerte wiederum aktivieren oder deaktivieren das Versenden der Reports. Es ist also theoretisch möglich mithilfe der Client-Methoden: „GetDataValuesReq“ und „SetDataValuesReq“ einen RCB zu konfigurieren und zu steuern. Dies erfordert eine genauere Kenntnis der Funktionsweise der RCBs da nicht auf jedes Attribut zur jeder Zeit zugegriffen werden darf. Einige Attributwerte erlauben nur einen Lesezugriff, andere wiederum auch einen Schreibzugriff. Einige Attributwerte erlauben einen Schreibzugriff nur dann, wenn sich der RCB in einem bestimmten Zustand befindet. Die Befehle zur Generalabfrage (GI) dürfen nur ausgeführt werden, wenn sich der RCB im aktivierten Zustand befindet. Um die Handhabung zu vereinfachen, bietet die TwinCAT IEC 61850 Implementierung noch weitere Möglichkeiten die RCBs zu konfigurieren und zu steuern. Im Folgenden werden alle verfügbaren Möglichkeiten (Methoden) kurz beschrieben.

Grundsätzlich werden die RCB-Attribute mit Initialwerten, die in TwinCAT Telecontrol Configurator für TrgOps, OptFlds, IntgPd, RptID, DataSet usw. vorkonfiguriert wurden, nach dem Download des SPS-Programms (SPS läuft noch nicht oder kein Bootprojekt aktiv) initialisiert. Wenn die Konfiguration aus einer ICD-Datei stammt, dann übernimmt TwinCAT Telecontrol Configurator die Initialwerte aus der ICD-Datei. Die Client-Projekte sind so konstruiert, dass sie nach dem Start der SPS zuerst eine Verbindung zum Server aufbauen und dann alle Server-Daten in den Client einlesen. Danach befinden sich auf der Client-Seite die Datenwerte, die auch auf dem Server vorhanden sind. Wenn die Server-Daten von den Initialwerten auf der Client-Seite abweichen, dann werden sie mit den Server-Daten überschrieben. Dazu gehören auch die Attribute der RCBs. Das Einlesen aller Werte wird in den Client-Projekten über zwei booleschen Variablen gesteuert: „bGetAllServerValues“ und „_bReadAllData“. Wenn „_bReadAllData“ TRUE ist dann werden alle Server-Daten immer einmalig nach der Herstellung oder Wiederherstellung der Verbindung eingelesen. Über eine steigende Flanke an „bGetAllServerValues“ können auch zum beliebigen Zeitpunkt die Werte erneut eingelesen werden. Sind beide booleschen Variablen auf FALSE gesetzt, werden die Daten nicht abgeglichen. Dies hängt von der Applikation ab.

Grundsätzlich ist festzuhalten, dass es sinnvoll ist die Client-Daten mit dem Server nach dem Verbindungsaufbau abzugleichen. Die Initialwerte, die im TwinCAT Telecontrol Configurator konfiguriert wurden, sind aber nicht verloren. Die Applikation kann auf diese Werte weiterhin über die Config-Eigenschaften der Datenattribute zugreifen. Jedes Datenattribut besitzt eine Eigenschaft: eConfig oder iConfig oder nConfig, bConfig (abhängig vom Basistyp des Datenattributes) usw. mit dem ursprünglichen Initialwert. In diesem Beispielprojekt werden unter anderem auch Initialwerte, die im TwinCAT Telecontrol Configurator konfiguriert wurden bei der Konfiguration und Aktivierung der RCBs verwendet.

Die in diesem Beispiel verwendeten RCB-Instanzen hängen an der Instanz des Logischen-Knoten: LLN0. Dort befinden sich auch z.B. die vom TwinCAT Telecontrol Configurator exportierten Konfigurationswerte für Report-Triggeroptionen (TrgOps) oder optionalen Report-Datenfelder (OptFlds).

```
FUNCTION_BLOCK FB_LN_IED_LD1_LLNO EXTENDS FB_AcsiCommonLogicalNodeClass
VAR_INPUT
    NamPlt: FB_DO_IED_LD1_LLNO_NamPlt := (sClass:='LPL', bLinkResult:=THIS^.AddDataToContainer(ipData:=NamPlt));
    Beh: FB_DO_IED_LD1_LLNO_Beh := (sClass:='ENS', bLinkResult:=THIS^.AddDataToContainer(ipData:=Beh));
    Health: FB_DO_IED_LD1_LLNO_Health := (sClass:='ENS', bLinkResult:=THIS^.AddDataToContainer(ipData:=Health));
    Mod: FB_DO_IED_LD1_LLNO_Mod := (sObjectName:='Mod', sClass:='ENC', bLinkResult:=THIS^.AddDataToContainer(ipData:=Mod));

    DS1: FB_DS_IED_LD1_LLNO_DS1 := (bLinkResult:=THIS^.AddDataSetToContainer(ipDataSet:=DS1));
    DS2: FB_DS_IED_LD1_LLNO_DS2 := (bLinkResult:=THIS^.AddDataSetToContainer(ipDataSet:=DS2));
    DS3: FB_DS_IED_LD1_LLNO_DS3 := (bLinkResult:=THIS^.AddDataSetToContainer(ipDataSet:=DS3));
```

```

    urcb101: FB_ScsmUrCBImpClass := (RptID:=(sValue:='IEDLD1/LLN0.urcb101'),
                                   DatSet:=(sValue:='IEDLD1/LLN0.DS1'),
                                   ConfRev:=(nValue:=1),
                                   OptFlds:=(SequenceNumber:=TRUE, ReportTimeStamp:=TRUE, ReasonFor
Inclusion:=TRUE, DataSetName:=TRUE, DataReference:=TRUE, BufferOverflow:=FALSE, EntryID:=FALSE, Conf
Revision:=TRUE),
                                   TrgOps:=(DataChange:=TRUE, QualityChange:=TRUE, DataUpdate:=FALS
E, Integrity:=TRUE, GeneralInterrogation:=TRUE),
                                   IntgPd:=(nValue:=2000),
                                   bLinkResult:=THIS^.AddUnbufferedReportControlBlockToContainer(ip
UnbufferedReportControlBlock:=urcb101));
    urcb201: FB_ScsmUrCBImpClass := (RptID:=(sValue:='IEDLD1/LLN0.urcb201'),
                                   DatSet:=(sValue:='IEDLD1/LLN0.DS2'),
                                   ConfRev:=(nValue:=1),
                                   OptFlds:=(SequenceNumber:=TRUE, ReportTimeStamp:=TRUE, ReasonFor
Inclusion:=TRUE, DataSetName:=TRUE, DataReference:=FALSE, BufferOverflow:=FALSE, EntryID:=FALSE, Con
fRevision:=TRUE),
                                   TrgOps:=(DataChange:=TRUE, QualityChange:=TRUE, DataUpdate:=TRUE
, Integrity:=TRUE, GeneralInterrogation:=TRUE),
                                   IntgPd:=(nValue:=5000),
                                   bLinkResult:=THIS^.AddUnbufferedReportControlBlockToContainer(ip
UnbufferedReportControlBlock:=urcb201));
    urcb301: FB_ScsmUrCBImpClass := (RptID:=(sValue:='IEDLD1/LLN0.urcb301'),
                                   DatSet:=(sValue:='IEDLD1/LLN0.DS3'),
                                   ConfRev:=(nValue:=1), OptFlds:=(SequenceNumber:=TRUE, ReportTime
Stamp:=TRUE, ReasonForInclusion:=TRUE, DataSetName:=TRUE, DataReference:=FALSE, BufferOverflow:=FALS
E, EntryID:=FALSE, ConfRevision:=TRUE),
                                   TrgOps:=(DataChange:=TRUE, QualityChange:=TRUE, DataUpdate:=TRUE
, Integrity:=TRUE, GeneralInterrogation:=TRUE),
                                   IntgPd:=(nValue:=5000),
                                   bLinkResult:=THIS^.AddUnbufferedReportControlBlockToContainer(ip
UnbufferedReportControlBlock:=urcb301));
    brcb101: FB_ScsmBrCBImpClass := (RptID:=(sValue:='IEDLD1/LLN0.brcb101'),
                                   DatSet:=(sValue:='IEDLD1/LLN0.DS1'),
                                   ConfRev:=(nValue:=1),
                                   OptFlds:=(SequenceNumber:=TRUE, ReportTimeStamp:=TRUE, ReasonFor
Inclusion:=TRUE, DataSetName:=TRUE, DataReference:=TRUE, BufferOverflow:=TRUE, EntryID:=TRUE, ConfRe
vision:=TRUE),
                                   TrgOps:=(DataChange:=TRUE, QualityChange:=TRUE, DataUpdate:=FALS
E, Integrity:=TRUE, GeneralInterrogation:=TRUE),
                                   IntgPd:=(nValue:=2000),
                                   bLinkResult:=THIS^.AddBufferedReportControlBlockToContainer(ipBu
fferedReportControlBlock:=brcb101));
    brcb201: FB_ScsmBrCBImpClass := (RptID:=(sValue:='IEDLD1/LLN0.brcb201'),
                                   DatSet:=(sValue:='IEDLD1/LLN0.DS2'),
                                   ConfRev:=(nValue:=1), OptFlds:=(SequenceNumber:=TRUE, ReportTime
Stamp:=TRUE, ReasonForInclusion:=TRUE, DataSetName:=TRUE, DataReference:=FALSE, BufferOverflow:=TRUE
, EntryID:=TRUE, ConfRevision:=TRUE),
                                   TrgOps:=(DataChange:=TRUE, QualityChange:=TRUE, DataUpdate:=TRUE
, Integrity:=TRUE, GeneralInterrogation:=TRUE),
                                   IntgPd:=(nValue:=5000),
                                   bLinkResult:=THIS^.AddBufferedReportControlBlockToContainer(ipBu
fferedReportControlBlock:=brcb201));
    brcb301: FB_ScsmBrCBImpClass := (RptID:=(sValue:='IEDLD1/LLN0.brcb301'),
                                   DatSet:=(sValue:='IEDLD1/LLN0.DS3'),
                                   ConfRev:=(nValue:=1),
                                   OptFlds:=(SequenceNumber:=TRUE, ReportTimeStamp:=TRUE, ReasonFor
Inclusion:=TRUE, DataSetName:=TRUE, DataReference:=FALSE, BufferOverflow:=TRUE, EntryID:=TRUE, ConfR
evision:=TRUE),
                                   TrgOps:=(DataChange:=TRUE, QualityChange:=TRUE, DataUpdate:=TRUE
, Integrity:=TRUE, GeneralInterrogation:=TRUE),
                                   IntgPd:=(nValue:=5000),
                                   bLinkResult:=THIS^.AddBufferedReportControlBlockToContainer(ipBu
fferedReportControlBlock:=brcb301));
END_VAR
VAR
END_VAR

```

Report-Konfiguration/-Steuerung mit den Methoden: GetDataValuesReq, SetDataValuesReq

Dies sind die Basismethoden um die RCBs zu konfigurieren und zu steuern. Sie bieten die volle Kontrolle über das Verhalten des RCBs, erfordern aber genauere Kenntnis der Funktionsweise der RCBs und mehr Programmieraufwand. Sie müssen eventuell eine komplexere Zustandsmaschine selbst implementieren in der Sie mehrere Schreib-/Lesezugriffe auf diverse RCB-Attributwerte mit Hilfe dieser Methoden durchführen. Verwenden Sie diese Methoden z. B. dann, wenn Sie ein Verhalten benötigen, welches dem Standard nicht

entspricht. In diesem Beispiel wird auf die Verwendung dieser Methoden zur Konfiguration und Steuerung der RCBs nicht weiter näher eingegangen. Folgende Schritte sind aber z. B. notwendig, um das Versenden der Reports eines ungepufferten RCBs zu aktivieren:

- Aktuelle Konfiguration und Statusinformation des RCBs lesen. Es werden dabei die Werte aller Attribute: „RptEna“, „Resv“, „OptFlds“, „DatSet“, „TrgOps“, ..., usw. mit der Methode: „GetDataValuesReq“ gelesen.
- Falls nötig neue Konfigurationswerte schreiben. Es werden dabei neue Werte mit der Methode: „SetDataValuesReq“ in die Attribute „OptFlds“, „DatSet“ oder „TrgOps“, ... usw. geschrieben.
- Den Wert: „TRUE“ ins Attribut: „Resv“ schreiben. Der RCB wird für die Verwendung mit dem Client reserviert.
- Den Wert: „TRUE“ ins Attribut: „RptEna“ schreiben. Das Versenden der Reports wird daraufhin aktiviert.

Folgende Schritte sind z. B. notwendig, um das Versenden der Reports eines ungepufferten RCBs zu deaktivieren:

- Den Wert: „FALSE“ ins Attribut: „RptEna“ schreiben. Das Versenden der Reports wird deaktiviert.
- Den Wert: „FALSE“ ins Attribut „Resv“ schreiben. Die Reservierung für die Verwendung des RCBs mit dem Client wird aufgehoben.

Report-Konfiguration/-Steuerung mit den Methoden: GetUrCBValuesReq, SetUrCBValuesReq, GetBrCBValuesReq, SetBrCBValuesReq

Diese Methoden ermöglichen das Lesen aller oder das Schreiben eines oder mehreren Attributwerte mit einem einzigen Methodenaufruf. Die Methoden: „GetUrCBValuesReq“ oder „GetBrCBValuesReq“ lesen alle Attributwerte eines RCBs. In diesem Fall werden beim Erfolg alle Attributwerte eines RCBs vom Server zum Client übertragen und in das TwinCAT IEC 61850-Datenmodell kopiert.

Die Methoden: „SetUrCBValuesReq“ und „SetBrCBValuesReq“ erlauben das Schreiben von einem oder mehreren Attributwerten eines RCBs. Beim Aktivieren/Deaktivieren der Reports spielt die Reihenfolge, in der die Attributwerte geschrieben werden, eine wichtige Rolle. Die meisten RCB-Konfigurationswerte können nur geschrieben werden, wenn die Reports nicht aktiv sind. Bei ungepufferten Reports muss z. B. der Attributwert „Resv“ zuerst auf „TRUE“ gesetzt werden, bevor der Attributwert „RptEna“ auf „TRUE“ gesetzt werden kann (um die Reports zu aktivieren). Beim Deaktivieren der Reports müssen diese Attributwerte in der umgekehrten Reihenfolge auf „FALSE“ gesetzt werden. D.h. zuerst „RptEna“ und dann „Resv“. Der Set-Methodenparameter: „stSet“ bestimmt welche Attributwerte geschrieben werden sollen. Die neuen zu schreibenden Werte werden zuerst in das Datenmodell reingeschrieben, dann die entsprechenden Attribute über den „stSet“-Parameter ausgewählt und die Set-Methode aufgerufen. Beim Erfolg werden die Daten vom IEC 61850 Client-Datenmodell zum Server übertragen.

Report-Konfiguration/-Steuerung mit den Methoden des RCB-Client-Objekts

Die einfachste Art die gepufferten oder ungepufferten Reports zu konfigurieren und zu steuern ist die Verwendung des Client-Funktionsbausteins unterhalb der Berichtssteuerblockinstanz. Der Client-Funktionsbaustein besitzt Eigenschaften, um den RCB zu konfigurieren und Methoden, um die Übertragung der Reports zu aktivieren oder zu deaktivieren. Im Bild unten zeigt z. B. das IntelliSense die verfügbaren Methoden und Eigenschaften bei einem gepufferten RCB:

```

-> ELSIF eControl_IEDLD1_LLNO_brcb101.<>.E_AcsiCtlReport.None THEN
-> IF eControl_IEDLD1_LLNO_brcb101.E_AcsiCtlReport.Enable THEN (*Example=>enable-reporting*)
-> bSuccess:=fbIED_IEDLD1_LLNO_brcb101.Client.EnableReq(ipClient:=fbConnection, ipDataSet:=fbIED_IEDLD1_LLNO_DS1, ipResult=>ipResult);
-> state:=SEL(bSuccess, *100, *11);
-> ELSIF eControl_IEDLD1_LLNO_brcb101.E_AcsiCtlReport.Equal THEN (*Example=>disable-reporting*)
-> bSuccess:=fbIED_IEDLD1_LLNO_brcb101.Client.EqualReq(ipClient:=fbConnection, ipDataSet:=fbIED_IEDLD1_LLNO_DS1, ipResult=>ipResult);
-> state:=SEL(bSuccess, *100, *11);
-> ELSIF eControl_IEDLD1_LLNO_brcb101.E_AcsiCtlReport.EqualEx THEN (*Example=>execute-general-interrogation*)
-> bSuccess:=fbIED_IEDLD1_LLNO_brcb101.Client.EqualExReq(ipClient:=fbConnection, ipDataSet:=fbIED_IEDLD1_LLNO_DS1, ipResult=>ipResult);
-> state:=SEL(bSuccess, *100, *11);
-> ELSIF eControl_IEDLD1_LLNO_brcb101.E_AcsiCtlReport.EqualTo THEN (*Example=>execute-purge-buffer*)
-> bSuccess:=fbIED_IEDLD1_LLNO_brcb101.Client.EqualToReq(ipClient:=fbConnection, ipDataSet:=fbIED_IEDLD1_LLNO_DS1, ipResult=>ipResult);
-> state:=SEL(bSuccess, *100, *11);
-> ELSIF eControl_IEDLD1_LLNO_brcb101.E_AcsiCtlReport.GIreq THEN (*Example=>execute-resync*)
-> bSuccess:=fbIED_IEDLD1_LLNO_brcb101.Client.GIreq(ipClient:=fbConnection, nEntryID:=Last_Rx_brcb101_nEntryID, ipResult=>ipResult);
-> state:=SEL(bSuccess, *100, *11);

```

Einige Methoden und Eigenschaften beim ungepufferten RCB:

```

→ → → ELSIF eControl_IEDLD1_LLNO_urcb101 <> E_AcsiCtlReport.None THEN
→ → → IF eControl_IEDLD1_LLNO_urcb101 = E_AcsiCtlReport.Enable THEN (*Example=>enable-reporting*)
→ → → → bSuccess := fbIED.IEDLD1_LLNO_urcb101.Client.EnableReq(ipClient:=fbConnection, ipDataSet:=fbIED.IEDLD1_LLNO_DS1, ipResult=>ipResult);
→ → → → state := SEL(bSuccess, *100, *11);
→ → → → ELSIF eControl_IEDLD1_LLNO_urcb101 = E_AcsiCtlReport.Disable THEN (*Example=>disable-reporting*)
→ → → → → bSuccess := fbIED.IEDLD1_LLNO_urcb101.Client.DisableReq(ipClient:=fbConnection, ipDataSet:=fbIED.IEDLD1_LLNO_DS1, ipResult=>ipResult);
→ → → → → state := SEL(bSuccess, *100, *11);
→ → → → ELSIF eControl_IEDLD1_LLNO_urcb101 = E_AcsiCtlReport.GeneralInterrogation THEN (*Example=>execute-general-interrogation*)
→ → → → → bSuccess := fbIED.IEDLD1_LLNO_urcb101.Client.GeneralInterrogation(ipClient:=fbConnection, ipDataSet:=fbIED.IEDLD1_LLNO_DS1, ipResult=>ipResult);
→ → → → → state := SEL(bSuccess, *100, *11);
→ → → → END_IF
→ → → eControl_IEDLD1_LLNO_urcb101 := E_AcsiCtlReport.None
→ → → (*=====Control*buf=====*)
→ → → ELSIF eControl_IEDLD1_LLNO_brcb101 <> E_AcsiCtlReport.None THEN
→ → → → IF eControl_IEDLD1_LLNO_brcb101 = E_AcsiCtlReport.Enable THEN (*Example=>enable-reporting*)
→ → → → → bSuccess := fbIED.IEDLD1_LLNO_brcb101.Client.EnableReq(ipClient:=fbConnection, ipDataSet:=fbIED.IEDLD1_LLNO_DS1, ipResult=>ipResult);
→ → → → → state := SEL(bSuccess, *100, *11);

```

Folgende Tabelle listet die wichtigsten Methoden und Eigenschaften des Client-Funktionsbausteins an der Berichtssteuerblockinstanz. Der Client-Funktionsbaustein arbeitet, wenn nicht anders konfiguriert mit vordefinierten Standardwerten für Triggeroptionen, optionalen Datenfeldern und max. Zeit zwischen den Integrity Reports. Die Standardwerte können durch Beschreiben der entsprechenden Eigenschaften bei jeder Client-Instanz individuell geändert werden. Die neuen Werte werden vor dem Aktivieren des Reports, wenn sie sich von den der Server-Seite unterscheiden, zuerst zum Server übertragen und erst dann wird der Report aktiviert.

| Methode/Eigen-schaft | Beschreibung | Initialwert |
|--|--|---|
| EnableReq | Aktiviert das Versenden der Reports | - |
| DisableReq | Deaktiviert (stoppt) das Versenden der Reports | - |
| GIReq | Aktiviert den Befehl zur Generalabfrage | - |
| PurgeBufReq (nur bei gepufferten RCBs) | Aktiviert den Befehl zum Zurücksetzen des Puffers | - |
| ResyncReq (nur bei gepufferten RCBs) | Aktiviert den Befehl zur Resynchronisation der gepufferten Daten | - |
| Clear | Setzt den Funktionsbaustein zurück | - |
| cOptFlds | Konfiguriert die im Report enthaltenen Datenfelder | Ungepufferte RCBs: Alle Datenfelder aktiviert außer der „DataReference“, „BufferOverflow“ und „EntryID“.
Gepufferte RCBs: Alle Datenfelder aktiviert außer der „DataReference“ |
| cTrgOps | Konfiguriert die Triggeroptionen für einen Report | Alle aktiviert („DataUpdate“, „DataChange“, „QualityChange“, „GeneralInterrogation“, „Integrity“) |
| iResvTms | OPTIONAL: Konfiguriert die Zeit für die Reservierung des RCBs für einen bestimmten Client. | 60 falls vorhanden |
| nBufTm | Max. Report-Pufferzeit in Millisekunden | 0 |
| nIntgPd | Max. Zeit zwischen den Integrity-Reports in Millisekunden | 5000 |
| sRptID | Report-ID | Leerstring. Ein neuer Wert, wenn gesetzt, wird nur dann zum Server übertragen, wenn er kein Leerstring ist und wenn er sich von dem aktuellen Serverwert unterscheidet. |

Beim Aktivieren einer Methode wird intern zuerst der aktuelle Status des RCBs vom Server abgefragt. Als nächstes werden dann, abhängig von der ausgewählten Methode, einige Eigenschaften zum Server übertragen. Die eingestellten Werte der Eigenschaften werden zum Server nur dann übertragen, wenn sich der Wert in der Eigenschaft zu dem jeweiligen Wert des entsprechenden Attributes im Datenmodell

unterscheidet. Zum Schluss werden intern die Statusinformationen erneut vom Server abgefragt. Einige Methoden benötigen weitere Parameter. Die Methode „EnableReq“ benötigt z. B. den Schnittstellenzeiger auf die Verbindungsinstanz und das DataSet.

Der Client-Funktionsbaustein besitzt eine strukturierte Ausgangsvariable mit dem Namen: „stInfo“. Diese Variable enthält RCB-Statusinformationen und Informationen über den zuletzt empfangenen Report. Diese Statusinformationen werden aber nur dann aktualisiert wenn auch der Client-Funktionsbaustein zur Konfiguration und Aktivierung der Reports verwendet wurde. Der Client-Funktionsbaustein aktualisiert seine Statusinformationen nicht wenn die anderen, oben beschriebenen Methoden (SetUrCBValuesReq, SetBrCBValuesReq oder SetDataValuesReq, usw.) zur Konfiguration oder Aktivierung der Reports verwendet werden.

| | | |
|----------------------|---------------------------|-----------------------|
| Client | FB_ScsmUrCBClientJobClass | |
| _eState | E_ASYNCSERVICESTATE | Completed |
| fbError | FB_ServiceErrorClass | |
| stInfo | ST_ScsmUrCBJobInfo | |
| eState | E_SCSMRCBSTATE | Enabled |
| sRptID | T_AcsiVisString129 | 'IEDLD1/LLN0.urcb101' |
| sDatSet | T_AcsiObjectReference | 'IEDLD1/LLN0.DS1' |
| nSqNum | BYTE | 33 |
| tTimeOfEntry | T_BinaryTime | |
| timeOfDay | TIME_OF_DAY | TOD#16:12:33.256 |
| day | WORD | 13864 |
| stReasonCode | ST_AcsiReasonCode | |
| DataChange | BIT | FALSE |
| QualityChange | BIT | FALSE |
| DataUpdate | BIT | FALSE |
| Integrity | BIT | TRUE |
| GeneralInterrogation | BIT | FALSE |
| ApplicationTrigger | BIT | FALSE |
| ipAA | I_ScsmAssociationClass | 16#FFFFB08F37741F00 |
| sRptID | T_AcsiVisString129 | " |
| cTrgOps | ST_AcsiTriggerConditions | |
| cOptFlds | ST_AcsiOptionalFields | |
| _nIntgPd | DWORD | 5000 |
| _nBufTm | DWORD | 0 |
| _bOwnedResv | BOOL | TRUE |

Im Online-Mode können die Statusinformationen des Client-Funktionsbausteins zur einfachen Diagnose verwendet werden.

Parameterliste zur Konfiguration der Standardwerte des RCB-Client-Objekts

Die Standardwerte aller Client-Instanzen können systemweit, falls es nötig ist durch Änderung in der Parameterliste: Param_Scsm (in der Tc3_iec61850_8_1 SPS-Bibliothek) geändert werden.

```

VAR_GLOBAL CONSTANT
  cBrCB_OptFlds : ST_AcsiOptionalFields:=(SequenceNumber:=1, ReportTimeStamp:=1, ReasonForInclusion:=1, DataSetName:=1, DataReference:=0, BufferOverflow:=1, EntryID:=1, ConfRevision:=1, Segmentation:=0);
  cBrCB_TrgOps : ST_AcsiTriggerConditions:=(DataChange:=1, QualityChange:=1, DataUpdate:=1, Integrity:=1, GeneralInterrogation:=1);
  cBrCB_IntgPd : DWORD:=5000;

  cUrCB_OptFlds : ST_AcsiOptionalFields:=(SequenceNumber:=1, ReportTimeStamp:=1, ReasonForInclusion:=1, DataSetName:=1, DataReference:=0, BufferOverflow:=0, EntryID:=0, ConfRevision:=1, Segmentation:=0);
  cUrCB_TrgOps : ST_AcsiTriggerConditions:=(DataChange:=1, QualityChange:=1, DataUpdate:=1, Integrity:=1, GeneralInterrogation:=1);
  cUrCB_IntgPd : DWORD:=5000;
  cBrCB_iResvTms : INT(-1..3600):=60;
END_VAR

```

Beispielprojekt

Der ungepufferte „urcb101“ und der gepufferte RCB „brcb101“ kann im Beispielprojekt auf zwei Arten gesteuert werden. Einmal mit Hilfe der Methoden: „GetUrCBValuesReq“, „GetBrCBValuesReq“, „SetUrCBValuesReq“ und „SetBrCBValuesReq“. Bei der anderen Möglichkeit handelt es sich um die Methoden am Client-Funktionsbaustein unterhalb der Berichtssteuerblockinstanz. Die weiter unten beschriebenen booleschen und Enum-Variablen finden Sie in der Implementierung des Funktionsbausteins: „FB_IEDClient“. Im Projektarchiv befindet sich auch eine ICD-Datei. Diese Datei beschreibt das im Beispielprojekt verwendete IEC 61850 Datenmodell. Sie können diese Datei z. B. in einer Drittherstellersoftware verwenden und einen Server simulieren.

Test mit Get-/Set-Methoden

Eine steigende Flanke an der booleschen Variablen: „bGetUrCBValues_urcb101“ oder „bGetBrCBValues_brbc101“ aktiviert das Lesen aller Attributwerte des RCB: „urcb101“ bzw. „brcb101“. Durch eine Wertänderung an der Aufzählungsvariablen „eSetUrCBValues_urcb101_Resv“ und „eSetUrCBValues_urcb101_RptEna“ oder „eSetBrCBValues_brbc101_RptEna“ kann der boolesche Wert der Datenattribute „Resv“ oder „RptEna“ auf „True“ oder „False“ gesetzt werden. Im Idealfall, wenn alle anderen RCB-Attributwerte bereits vorkonfiguriert und diese Werte sich bereits auf der Serverseite befinden, kann der RCB: „urcb101“ auf folgende Weise aktiviert werden:
Der Wert „Enable“ geschrieben in die Variable: „eSetUrCBValues_urcb101_Resv“ beschreibt das Attribut: „Resv“ mit dem Wert „TRUE“. Der Wert „Enable“ geschrieben in die Variable: „eSetUrCBValues_urcb101_RptEna“ beschreibt das Attribut: „RptEna“ mit dem Wert „TRUE“. Danach ist die Übertragung der ungepufferten Reports von „urcb101“ aktiviert.

Der RCB: „brcb101“ kann auf folgende Weise aktiviert werden:

Der Wert „Enable“ geschrieben in die Variablen: „eSetBrCBValues_brbc101_RptEna“ beschreibt das Attribut: „RptEna“ mit dem Wert „TRUE“. Danach ist die Übertragung der gepufferten Reports von „brcb101“ aktiviert.

```

...
(*=====
=====*)
(* Write "IEDLD1/LLN0.urcb101.Resv" data attribute value: True|False *)
ELSIF eSetUrCBValues_urcb101_Resv <> E_CtlBool.None THEN (* Example => execute SetUrCBValuesReq
() command *)
    memset(ADR(stSet_urcb), 0, SIZEOF(stSet_urcb)); (* Clear all URcb attribute selection flags
*)
    stSet_urcb.Resv:= TRUE; (* Select URcb->Resv attribute to be written *)
    fbIED.IEDLD1.LLN0.urcb101.Resv.bValue:= SEL(eSetUrCBValues_urcb101_Resv = E_CtlBool.Enable,
FALSE, TRUE); (* Set new data attribute value to be written *)
    bSuccess:= fbConnection.SetUrCBValuesReq(ipURcb:=fbIED.IEDLD1.LLN0.urcb101, stSet:=stSet_ur
cb, hUser:=0, ipSink:=0, nInvokeID=>nInvokeID, ipResult=>ipResult); (* Activation of command executi
on *)
    state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command co
mpletion else report an error *)
    eSetUrCBValues_urcb101_Resv:= E_CtlBool.None; (* Reset enum value to recognize next write r
equest *)

(*=====
=====*)
(* Write "IEDLD1/LLN0.urcb101.RptEna" data attribute value: True|False *)
ELSIF eSetUrCBValues_urcb101_RptEna <> E_CtlBool.None THEN (* Example => execute SetUrCBValuesR
eq() command *)
    memset(ADR(stSet_urcb), 0, SIZEOF(stSet_urcb)); (* Clear all URcb attribute selection flags
*)
    stSet_urcb.RptEna:= TRUE; (* Select URcb->RptEna attribute to be written *)
    fbIED.IEDLD1.LLN0.urcb101.RptEna.bValue:= SEL(eSetUrCBValues_urcb101_RptEna = E_CtlBool.Ena
ble, FALSE, TRUE); (* Set new data attribute value to be written *)
    bSuccess:= fbConnection.SetUrCBValuesReq(ipURcb:=fbIED.IEDLD1.LLN0.urcb101, stSet:=stSet_ur
cb, hUser:=0, ipSink:=0, nInvokeID=>nInvokeID, ipResult=>ipResult); (* Activation of command executi
on *)
    state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command co
mpletion else report an error *)
    eSetUrCBValues_urcb101_RptEna:= E_CtlBool.None; (* Reset enum value to recognize next write
request *)

...

(*=====
=====*)
(* Write "IEDLD1/LLN0.brbc101.RptEna" data attribute value *)
ELSIF eSetBrCBValues_brbc101_RptEna <> E_CtlBool.None THEN (* Example => execute SetBrCBValuesR
eq() command *)

```

```

memset(ADR(stSet_brcb), 0, sizeof(stSet_brcb)); (* Clear all BRCB attribute selection flags
*)
stSet_brcb.RptEna:= TRUE; (* Select BRCB->RptEna attribute to be written *)
fbIED.IEDLD1.LLN0.brcb101.RptEna.bValue:= SEL(eSetBrCBValues_brcb101_RptEna = E_CtlBool.Ena
ble, FALSE, TRUE); (* Set new data attribute value to be written *)
bSuccess:= fbConnection.SetBrCBValuesReq(ipBrCB:=fbIED.IEDLD1.LLN0.brcb101, stSet:=stSet_br
cb, hUser:=0, ipSink:=0, nInvokeID=>nInvokeID, ipResult=>ipResult); (* Activation of command executi
on *)
state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command co
mpletion else report an error *)
eSetBrCBValues_brcb101_RptEna:= E_CtlBool.None; (* Reset enum value to recognize next write
request *)
...

```

Um die Übertragung der ungepufferten Reports zu stoppen, muss zuerst der Wert „Disable“ in die Variable: „eSetUrCBValues_urcb101_RptEna“ und dann in die Variable: „eSetUrCBValues_urcb101_Resv“ geschrieben werden. Dies setzt die Werte der Datenattribute „RptEna“ und „Resv“ auf FALSE.

Um die Übertragung der gepufferten Reports zu stoppen, muss der Wert „Disable“ in die Variable „eSetBrCBValues_brcb101_RptEna“ geschrieben werden. Dies setzt den Wert des Datenattributs „RptEna“ auf FALSE.

Test mit Client-Methoden an der Berichtssteuerblockinstanz

Um sämtliche Zustände eines RCBs zu berücksichtigen, wird hier ein Enum zur Aktivierung von „urcb101“ bzw. „brcb101“ Steuerbefehlen verwendet. Sobald das Enum „eControl_IEDLD1_LLN0_urcb101“ oder „eControl_IEDLD1_LLN0_brcb101“ nicht „E_AcsiCtlReport.None“ ist, wird der entsprechende Befehl am RCB aktiviert.

Anhand des „ipResult“-Ausgangs des Funktionsbausteins kann der Zustand der Abarbeitung des letzten Befehls abgefragt werden. Da die Abarbeitung der Methoden länger als einen Zyklus benötigt, wird die State-Machine in den State 11 versetzt. Dort muss die Methode „Execute“ des Schnittstellenzeigers „ipResult“ so lange aufgerufen werden bis „ipResult.IsBusy()“ den Wert „FALSE“ liefert. Falls dieser sich in der Bearbeitung der Funktionalität befindet, wird „IsBusy()“ als „TRUE“ zurückgegeben und die State-Machine des verweilt weiterhin im State 11. Sobald der Befehl erfolgreich abgearbeitet wurde, wird die State-Machine in den State 0 versetzt. Die Instanzen „urcb101“ und „brcb101“ werden in diesem Beispiel mit den Werten für „TrgOps“, „OptFlds“ und „IntgPd“ aktiviert, die im TwinCAT Telecontrol Configurator konfiguriert wurden.

```

...
(*=====*)
(* Control unbuffered reports using "IEDLD1/LLN0.urcb101.Client" function block instance *)
ELSIF eControl_IEDLD1_LLN0_urcb101 <> E_AcsiCtlReport.None THEN
  IF eControl_IEDLD1_LLN0_urcb101 = E_AcsiCtlReport.Enable THEN (* Example => enable reporting *)

    (* Changing default properties of "Client" function block configures reporting behaviour and
    content of report message data.
    The "Client" function block writes the new property values to the server and enables RCB.
    In this example we use configuration values that were defined in the TwinCAT Telecontrol
    Configurator to control "urcb101".*)

    (* Configures "RptID" (report ID) to be used in report message *)
    fbIED.IEDLD1.LLN0.urcb101.Client.sRptID:= fbIED.IEDLD1.LLN0.urcb101.RptID.sConfig;
    (* => "IEDLD1/.LLN0.urcb101" *)

    (* Configures "TrgOps" (trigger options) to be used to trigger new report messages *)
    fbIED.IEDLD1.LLN0.urcb101.Client.cTrgOps:= fbIED.IEDLD1.LLN0.urcb101.TrigOps.cConfig;
    (* => All enabled except "DataUpdate" *)

    (* Configures "OptFlds" (optional fields) to be used in report message *)
    fbIED.IEDLD1.LLN0.urcb101.Client.cOptFlds:= fbIED.IEDLD1.LLN0.urcb101.OptFlds.cConfig;
    (* => All enabled except "BufferOverflow" and "EntryID" *)

    (* Configures "IntgPd" (period in milliseconds used to generate integrity report) *)
    fbIED.IEDLD1.LLN0.urcb101.Client.nIntgPd:= fbIED.IEDLD1.LLN0.urcb101.IntgPd.nConfig;
    (* => 2000ms *)

    (* Configures "BufTm" (max. report message buffer time in milliseconds) *)
    fbIED.IEDLD1.LLN0.urcb101.Client.nBufTm:= fbIED.IEDLD1.LLN0.urcb101.BufTm.nConfig;
    (* => 0ms *)

    (* Write properties and enable RCB *)
    bSuccess:= fbIED.IEDLD1.LLN0.urcb101.Client.EnableReq(ipClient:=fbConnection, ipDataSet:=fbI

```

```

ED.IEDLD1.LLN0.DS1, ipResult=>ipResult); (* Activation of command execution *)
    state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command completion else report an error *)
    ELSIF eControl_IEDLD1_LLN0_urcb101 = E_AcsiCtlReport.Disable THEN (* Example => disable reporting *)
        bSuccess:= fbIED.IEDLD1.LLN0.urcb101.Client.DisableReq(ipResult=>ipResult); (* Activation of command execution *)
        state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command completion else report an error *)
    ELSIF eControl_IEDLD1_LLN0_urcb101 = E_AcsiCtlReport.GI THEN (* Example => execute general interrogation *)
        bSuccess:= fbIED.IEDLD1.LLN0.urcb101.Client.GIReq(ipResult=>ipResult); (* Activation of command execution *)
        state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command completion else report an error *)
    END_IF
    eControl_IEDLD1_LLN0_urcb101:= E_AcsiCtlReport.None; (* Reset enum value to recognize next write request *)
...
(*=====*)
(* Control buffered reports using "IEDLD1/LLN0.brCb101.Client" function block instance *)
ELSIF eControl_IEDLD1_LLN0_brCb101 <> E_AcsiCtlReport.None THEN
    IF eControl_IEDLD1_LLN0_brCb101 = E_AcsiCtlReport.Enable THEN (* Example => enable reporting using *)
        (*
        Changing default properties of "Client" function block configures reporting behaviour and content of report message data.
        The "Client" function block writes the new property values to the server and enables RCB.
        In this example we use configuration values that were defined in the TwinCAT Telecontrol Configurator to control "brCb101".
        *)
        (* Configures "RptID" to be used in report message *)
        fbIED.IEDLD1.LLN0.brCb101.Client.sRptID:= fbIED.IEDLD1.LLN0.brCb101.RptID.sConfig;
(* => "IEDLD1/LLN0.brCb101" *)

        (* Configures "TrgOps" (trigger options) to be used to trigger new report messages *)
        fbIED.IEDLD1.LLN0.brCb101.Client.cTrgOps:= fbIED.IEDLD1.LLN0.brCb101.TrgOps.cConfig;
(* => All enabled except "DataUpdate" *)

        (* Configures "OptFlds" (option fields) to be used in report message *)
        fbIED.IEDLD1.LLN0.brCb101.Client.cOptFlds:= fbIED.IEDLD1.LLN0.brCb101.OptFlds.cConfig;
(* => All enabled *)

        (* Configures "IntgPd" (period in milliseconds used to generate integrity report) *)
        fbIED.IEDLD1.LLN0.brCb101.Client.nIntgPd:= fbIED.IEDLD1.LLN0.brCb101.IntgPd.nConfig;
(* => 2000ms *)

        (* Configures "BufTm" (max. report message buffer time in milliseconds) *)
        fbIED.IEDLD1.LLN0.brCb101.Client.nBufTm:= fbIED.IEDLD1.LLN0.brCb101.BufTm.nConfig;
(* => 0ms *)

        (* Write properties and enable RCB *)
        bSuccess:= fbIED.IEDLD1.LLN0.brCb101.Client.EnableReq(ipClient:=fbConnection, ipDataSet:=fbIED.IEDLD1.LLN0.DS1, ipResult=>ipResult); (* Activation of command execution *)
        state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command completion else report an error *)
    ELSIF eControl_IEDLD1_LLN0_brCb101 = E_AcsiCtlReport.Disable THEN (* Example => disable reporting *)
        bSuccess:= fbIED.IEDLD1.LLN0.brCb101.Client.DisableReq(ipResult=>ipResult); (* Activation of command execution *)
        state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command completion else report an error *)
    ELSIF eControl_IEDLD1_LLN0_brCb101 = E_AcsiCtlReport.GI THEN (* Example => execute general interrogation *)
        bSuccess:= fbIED.IEDLD1.LLN0.brCb101.Client.GIReq(ipResult=>ipResult); (* Activation of command execution *)
        state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command completion else report an error *)
    ELSIF eControl_IEDLD1_LLN0_brCb101 = E_AcsiCtlReport.PurgeBuf THEN (* Example => execute purge buffer *)
        bSuccess:= fbIED.IEDLD1.LLN0.brCb101.Client.PurgeBufReq(ipClient:=fbConnection, ipResult=>ipResult); (* Activation of command execution *)
        state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command completion else report an error *)
    ELSIF eControl_IEDLD1_LLN0_brCb101 = E_AcsiCtlReport.Resync THEN (* Example => execute resync *)
        bSuccess:= fbIED.IEDLD1.LLN0.brCb101.Client.ResyncReq(ipClient:=fbConnection, nEntryID:=LastRx_brCb101_nEntryID, ipResult=>ipResult); (* Activation of command execution *)

```

```

state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command completion else report an error *)
END_IF
eControl_IEDLD1_LLNO_brcb101:= E_AcsiCtlReport.None;
(* Reset enum value to recognize next write request *)
...

```

Zu Demonstrationszwecken werden die Instanzen „urcb201“ und „brcb201“ dagegen nicht mit den im TwinCAT Telecontrol Configurator konfigurierten und exportierten Werten für „TrgOps“, „OptFlds“, „IntgPd“ usw. aktiviert. Stattdessen werden bei der Aktivierung des Reports die Standardwerte der „Client“-Instanz verwendet.

```

...
(*=====*)
(* Control unbuffered reports using "IEDLD1/LLNO.urcb201.Client" function block instance *)
ELSIF eControl_IEDLD1_LLNO_urcb201 <> E_AcsiCtlReport.None THEN
  IF eControl_IEDLD1_LLNO_urcb201 = E_AcsiCtlReport.Enable THEN (* Example => enable reporting *)
    (*
      Changing default properties of "Client" function block configures reporting behaviour and content of report message data (optional).
      The "Client" function block writes the new property values to the server and enables RCB.
      In this example we use the standard configuration values
of the function block "Client" to control the "urcb201":
      "fbIED.IEDLD1.LLN0.urcb201.Client.sRptID" := Empty string. Property "sRptID" is written if <
> '' and differs from server value.
      "fbIED.IEDLD1.LLN0.urcb201.Client.cTrgOps" := All trigger options are enabled
      "fbIED.IEDLD1.LLN0.urcb201.Client.cOptFlds" := All options are enabled except "DataReference", "BufferOverflow" and "EntryID"
      "fbIED.IEDLD1.LLN0.urcb201.Client.nIntgPd" := Set to 5000ms
      "fbIED.IEDLD1.LLN0.urcb201.Client.nBufTm" := Set to 0ms
    *)
    (* Write properties and enable RCB *)
    bSuccess:= fbIED.IEDLD1.LLN0.urcb201.Client.EnableReq(ipClient:=fbConnection, ipDataSet:=fbIED.IEDLD1.LLN0.DS2, ipResult=>ipResult); (* Activation of command execution *)
    state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command completion else report an error *)
    ELSIF eControl_IEDLD1_LLNO_urcb201 = E_AcsiCtlReport.Disable THEN (* Example => disable reporting *)
      bSuccess:= fbIED.IEDLD1.LLN0.urcb201.Client.DisableReq(ipResult=>ipResult); (* Activation of command execution *)
      state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command completion else report an error *)
    ELSIF eControl_IEDLD1_LLNO_urcb201 = E_AcsiCtlReport.GI THEN (* Example => execute general interrogation *)
      bSuccess:= fbIED.IEDLD1.LLN0.urcb201.Client.GIReq(ipResult=>ipResult); (* Activation of command execution *)
      state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command completion else report an error *)
    END_IF
    eControl_IEDLD1_LLNO_urcb201:= E_AcsiCtlReport.None; (* Reset enum value to recognize next write request *)
  ...
(*=====*)
(* Control buffered reports using "IEDLD1/LLNO.brcb201.Client" function block instance *)
ELSIF eControl_IEDLD1_LLNO_brcb201 <> E_AcsiCtlReport.None THEN
  IF eControl_IEDLD1_LLNO_brcb201 = E_AcsiCtlReport.Enable THEN (* Example => enable reporting *)
    (*
      Changing default properties of "Client" function block configures reporting behaviour and content of report message data (optional).
      The "Client" function block writes the new property values to the server and enables RCB.
      In this example we use the standard configuration values
of the function block "Client" to control the "brcb201":
      "fbIED.IEDLD1.LLN0.brcb201.Client.sRptID" := Empty string. Property "sRptID" is only written if <> '' and differs from server value
      "fbIED.IEDLD1.LLN0.brcb201.Client.cTrgOps" := All trigger options are enabled
      "fbIED.IEDLD1.LLN0.brcb201.Client.cOptFlds" := All options are enabled except "DataReference"
"
      "fbIED.IEDLD1.LLN0.brcb201.Client.nIntgPd" := Set to 5000ms
      "fbIED.IEDLD1.LLN0.brcb201.Client.nBufTm" := Set to 0ms
    *)
    (* Write properties and enable RCB *)
    bSuccess:= fbIED.IEDLD1.LLN0.brcb201.Client.EnableReq(ipClient:=fbConnection, ipDataSet:=fbIED.IEDLD1.LLN0.DS2, ipResult=>ipResult); (* Activation of command execution *)
  ...

```

```

state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command completion else report an error *)
ELSIF eControl_IEDLD1_LLNO_brcb201 = E_AcsiCtlReport.Disable THEN (* Example => disable reporting *)
    bSuccess:= fbIED.IEDLD1_LLNO_brcb201.Client.DisableReq(ipResult=>ipResult); (* Activation of command execution *)
    state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command completion else report an error *)
    ELSIF eControl_IEDLD1_LLNO_brcb201 = E_AcsiCtlReport.GI THEN (* Example => execute general interrogation *)
        bSuccess:= fbIED.IEDLD1_LLNO_brcb201.Client.GIreq(ipResult=>ipResult); (* Activation of command execution *)
        state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command completion else report an error *)
        ELSIF eControl_IEDLD1_LLNO_brcb201 = E_AcsiCtlReport.PurgeBuf THEN (* Example => execute purge buffer *)
            bSuccess:= fbIED.IEDLD1_LLNO_brcb201.Client.PurgeBufReq(ipClient:=fbConnection, ipResult=>ipResult); (* Activation of command execution *)
            state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command completion else report an error *)
            ELSIF eControl_IEDLD1_LLNO_brcb201 = E_AcsiCtlReport.Resync THEN (* Example => execute resync *)
                bSuccess:= fbIED.IEDLD1_LLNO_brcb201.Client.ResyncReq(ipClient:=fbConnection, nEntryID:=LastRx_brcb201_nEntryID, ipResult=>ipResult); (* Activation of command execution *)
                state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command completion else report an error *)
            END_IF
            eControl_IEDLD1_LLNO_brcb201:= E_AcsiCtlReport.None; (* Reset enum value to recognize next write request *)
...

```

Die Instanzen „urcb301“ und „brcb301“ werden wiederum mit benutzerdefinierten Werten für „TrgOps“, „OptFlds“ und „IntgPd“ aktiviert.

```

...
(*=====*)
(* Control unbuffered reports using "IEDLD1/LLNO.urcb301.Client" function block instance *)
ELSIF eControl_IEDLD1_LLNO_urcb301 <> E_AcsiCtlReport.None THEN
    IF eControl_IEDLD1_LLNO_urcb301 = E_AcsiCtlReport.Enable THEN (* Example => enable reporting *)

        (*
        Changing default properties of "Client" function block configures reporting behaviour and content of report message data.
        The "Client" function block writes the new property values to the server and enables RCB.
        In this example we use configuration values that were defined by application/user to control the "urcb301".
        *)

        (* Configures "RptID" (report ID) to be used in report message *)
        fbIED.IEDLD1_LLNO.urcb301.Client.sRptID:= 'IEDLD1/LLNO.RP.urcb301';

        (* Configures "TrgOps" (trigger options) to be used to trigger new report messages *)
        usrTrgOps.DataChange:= TRUE;
        usrTrgOps.DataUpdate:= FALSE;
        usrTrgOps.QualityChange:= FALSE;
        usrTrgOps.Integrity:= TRUE;
        usrTrgOps.GeneralInterrogation:= TRUE;
        fbIED.IEDLD1_LLNO.urcb301.Client.cTrgOps:= usrTrgOps;

        (* Configures "OptFlds" (option fields) to be used in report message *)
        usrOptFlds.ConfRevision:= TRUE;
        usrOptFlds.DataReference:= FALSE;
        usrOptFlds.DataSetName:= FALSE;
        usrOptFlds.ReasonForInclusion:= TRUE;
        usrOptFlds.ReportTimeStamp:= FALSE;
        usrOptFlds.SequenceNumber:= TRUE;
        usrOptFlds.BufferOverflow:= FALSE;
        usrOptFlds.EntryID:= FALSE;
        fbIED.IEDLD1_LLNO.urcb301.Client.cOptFlds:= usrOptFlds;

        (* Configures "IntgPd" (period in milliseconds used to generate integrity report) *)
        fbIED.IEDLD1_LLNO.urcb301.Client.nIntgPd:= 1000;

        (* Configures "BufTm" (max. report message buffer time in milliseconds) *)
        fbIED.IEDLD1_LLNO.urcb301.Client.nBufTm:= 500;

        (* Write properties and enable RCB *)
        bSuccess:= fbIED.IEDLD1_LLNO.urcb301.Client.EnableReq(ipClient:=fbConnection, ipDataSet:=fbIED.IEDLD1_LLNO.DS3, ipResult=>ipResult); (* Activation of command execution *)
...

```



```

state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command comp
letion else report an error *)
ELSIF eControl_IEDLD1_LLNO_urcb301 = E_AcsiCtlReport.Disable THEN (* Example => disable reportin
g *)
    bSuccess:= fbIED.IEDLD1.LLN0.urcb301.Client.DisableReq(ipResult=>ipResult); (* Activation of
command execution *)
    state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command comp
letion else report an error *)
    ELSIF eControl_IEDLD1_LLNO_urcb301 = E_AcsiCtlReport.GI THEN (* Example => execute general inter
rogation *)
        bSuccess:= fbIED.IEDLD1.LLN0.urcb301.Client.GIReq(ipResult=>ipResult); (* Activation of comma
nd execution *)
        state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command comp
letion else report an error *)
    END_IF
    eControl_IEDLD1_LLNO_urcb301:= E_AcsiCtlReport.None; (* Reset enum value to recognize next write
request *)
...
(*=====*)
(* Control buffered reports using "IEDLD1/LLNO.brbc301.Client" function block instance *)
ELSIF eControl_IEDLD1_LLNO_brbc301 <> E_AcsiCtlReport.None THEN
    IF eControl_IEDLD1_LLNO_brbc301 = E_AcsiCtlReport.Enable THEN (* Example => enable reporting usi
ng *)
        (*
        Changing default properties of "Client" function block configures reporting behaviour and c
ontent of report message data.
        The "Client" function block writes the new property values to the server and enables RCB.
        In this example we use configuration values that were defined by application/
user to control the "brbc301".
        *)
        (* Configures "RptID" (report ID) to be used in report message *)
        fbIED.IEDLD1.LLN0.brbc301.Client.sRptID:= 'IEDLD1/LLNO.BR.brbc301';

        (* Configures "TrgOps" (trigger options) to be used to trigger new report messages *)
        usrTrgOps.DataChange:= TRUE;
        usrTrgOps.DataUpdate:= FALSE;
        usrTrgOps.QualityChange:= FALSE;
        usrTrgOps.Integrity:= TRUE;
        usrTrgOps.GeneralInterrogation:= TRUE;
        fbIED.IEDLD1.LLN0.brbc301.Client.cTrgOps:= usrTrgOps;

        (* Configures "OptFlds" (option fields) to be used in report message *)
        usrOptFlds.ConfRevision:= TRUE;
        usrOptFlds.DataReference:= FALSE;
        usrOptFlds.DataSetName:= FALSE;
        usrOptFlds.ReasonForInclusion:= TRUE;
        usrOptFlds.ReportTimeStamp:= FALSE;
        usrOptFlds.SequenceNumber:= TRUE;
        usrOptFlds.BufferOverflow:= TRUE;
        usrOptFlds.EntryID:= TRUE;
        fbIED.IEDLD1.LLN0.brbc301.Client.cOptFlds:= usrOptFlds;

        (* Configures "IntgPd" (period in milliseconds used to generate integrity report) *)
        fbIED.IEDLD1.LLN0.brbc301.Client.nIntgPd:= 10000;

        (* Configures "BufTm" (max. report message buffer time in milliseconds) *)
        fbIED.IEDLD1.LLN0.brbc301.Client.nBufTm:= 500;

        (* Write properties and enable RCB *)
        bSuccess:= fbIED.IEDLD1.LLN0.brbc301.Client.EnableReq(ipClient:=fbConnection, ipDataSet:=fbIE
D.IEDLD1.LLN0.DS3, ipResult=>ipResult); (* Activation of command execution *)
        state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command comp
letion else report an error *)
        ELSIF eControl_IEDLD1_LLNO_brbc301 = E_AcsiCtlReport.Disable THEN (* Example => disable reportin
g *)
            bSuccess:= fbIED.IEDLD1.LLN0.brbc301.Client.DisableReq(ipResult=>ipResult); (* Activation of
command execution *)
            state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command comp
letion else report an error *)
            ELSIF eControl_IEDLD1_LLNO_brbc301 = E_AcsiCtlReport.GI THEN (* Example => execute general inter
rogation *)
                bSuccess:= fbIED.IEDLD1.LLN0.brbc301.Client.GIReq(ipResult=>ipResult); (* Activation of comma
nd execution *)
                state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command comp
letion else report an error *)
            ELSIF eControl_IEDLD1_LLNO_brbc301 = E_AcsiCtlReport.PurgeBuf THEN (* Example => execute purge b

```

```
uffer *)
    bSuccess:= fbIED.IEDLD1.LLN0.brcb301.Client.PurgeBufReq(ipClient:=fbConnection, ipResult=>ipR
esult); (* Activation of command execution *)
    state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command comp
letion else report an error *)
    ELSIF eControl_IEDLD1_LLN0_brcb301 = E_AcsiCtlReport.Resync THEN (* Example => execute resync *)
        bSuccess:= fbIED.IEDLD1.LLN0.brcb301.Client.ResyncReq(ipClient:=fbConnection, nEntryID:=Last_
Rx_brcb301_nEntryID, ipResult=>ipResult); (* Activation of command execution *)
        state:= SEL(bSuccess, 100, 11); (* If command activation succeeded then wait for command comp
letion else report an error *)
    END IF
    eControl_IEDLD1_LLN0_brcb301:= E_AcsiCtlReport.None; (* Reset enum value to recognize next write
request *)
...

```



Dies ist nur eine exemplarische Implementierung der Report Control Block-Funktionalität in der TwinCAT SPS.

7.6 Client - Direct Control with normal security

Dieses Beispiel zeigt die exemplarische Implementierung der Direktbefehle (ctlModel := 1, direct control with normal security) in einem TwinCAT IEC 61850 Client-Projekt für folgende Datenklassen der Schaltersteuerung (Common Data Class, CDC):

- SPC (controllable single point) object
- DPC (controllable double point) object
- APC (controllable analogue process value) object
- BAC (binary controlled analogue process value) object
- BSC (binary controlled step position information) object
- ENC (controllable enumerated status) object: Mod
- INC (controllable integer status) object
- ISC (integer controlled step position information) object

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1031/TF6510_TC3_IEC61850/Resources/5494787339.zip

Das hier beschriebene Beispiel nutzt die State Machine, welche in dem Kapitel „Allgemeine Client - Projektstruktur [► 591]“ beschrieben ist. Die States: 0, 1, 11 und 100 sind identisch zu der dort beschriebenen State Machine. Andere States wurden für das Beispiel modifiziert oder auch neue States hinzugefügt.

Für die Implementierung der direkten Befehle wird ein zusätzlicher, selbstimplementierter Funktionsbaustein benötigt. Im Beispiel wurde dafür ein neuer Funktionsbaustein: „FB_DirectControl“ hinzugefügt. Der Funktionsbaustein: „FB_DirectControl“ wickelt die eigentliche Befehlsausführung für die unterschiedlichen CDC-Typen ab. Im modifizierten State 10 vom „FB_IEDClient“ sind mehrere IF-Anweisungen aufgeführt, welche über die Instanz des „FB_DirectControl“-Funktionsbausteins die Befehlsausführung des Clients aktivieren. Im Deklarationsteil von „FB_IEDClient“ sind mehrere boolesche Variablen definiert. Eine steigende Flanke an der entsprechenden booleschen Variablen aktiviert in der IF-Anweisung die Befehlsausführung. Dies kann der Benutzer über die Online-Werte der Applikation auslösen.

Methoden des "FB_DirectControl"-Funktionsbausteins

| Methodenname | Beschreibung |
|--|--|
| APC
BAC
BSC
DPC
ENC_Mod
INC
ISC
SPC | Für jeden steuerbaren CDC-Typ ist eine Methode hinzugefügt worden, da die meisten CDCs unterschiedliche Kontrollwert-Typen verwenden, z. B. „APC“ -> „REAL“ und „SPC“ -> „BOOL“. Weil die CDCs unterschiedliche Typen verwenden, besitzt auch jede dieser Methoden für den Kontrollwert (z. B. „ctlVal“) einen anderen Eingangsparameter. Innerhalb der Methode wird der Wert auf einen Union-Typ geschrieben, welches für jeden CDC-Typ einen Member enthält. Die Union-Variable wird an die „Request“-Methode übergeben. Die „Request“-Methode ist eine private Methode und sie aktiviert die Befehlsausführung. |
| Execute | Sobald die „Request“-Methode des Funktionsbausteins erfolgreich aufgerufen wurde, muss die „Execute“-Methode in jeden Zyklus aufgerufen werden. Innerhalb dieser Methode befindet sich eine separate Statemachine, welche unabhängig von der Statemachine im Client-Funktionsbaustein arbeitet. Hierdurch wird der Code in kleinere Abschnitte aufgeteilt und kann in Zukunft besser erweitert werden. |
| Clear | Bricht den Befehl ab und setzt den Funktionsbaustein zurück. |

Der Befehl, abhängig vom CDC-Typen, modifiziert den Kontrollwert des CDCs. Zu Demonstrationszwecken wird, z. B. ein Zahlenwert hochgezählt oder ein boolescher Wert getoggelt.

Sobald einer der Befehle ausgelöst und erfolgreich abgearbeitet wurde, wird die Statemachine des „FB_IEDClient“-Funktionsbausteins in den State 11 versetzt. Hier verweilt die Statemachine, bis der Befehl erfolgreich durchgeführt wurde. Dafür wird in dem State 11 die „Execute“-Methode in jedem Zyklus aufgerufen und anhand des „ipResult“-Schnittstellenzeigers kann ermittelt werden, ob der Befehl noch bearbeitet wird oder bereits abgeschlossen ist. Dies wird über die „IsBusy()“-Methode abgefragt. Sobald der Befehl erfolgreich durchgeführt wurde, wechselt die Statemachine in den State 0.



Dies ist nur eine exemplarische Implementierung der Direktbefehle in der TwinCAT SPS.

7.7 Client - SBO Control with normal security

Dieses Beispiel zeigt die exemplarische Implementierung des Befehls: Anwahl-vor-Ausführung (ctlModel := 2, select before operate control with normal security) in einem TwinCAT IEC 61850 Client-Projekt für folgende Datenklassen der Schaltersteuerung (Common Data Class, CDC):

- SPC (controllable single point) object;
- DPC (controllable double point) object
- APC (controllable analogue process value) object
- BAC (binary controlled analogue process value) object
- BSC (binary controlled step position information) object
- ENC (controllable enumerated status) object: Mod
- INC (controllable integer status) object
- ISC (integer controlled step position information) object

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1031/TF6510_TC3_IEC61850/Resources/5494803595.zip

Das hier beschriebene Beispiel nutzt die Statemachine, welche in dem Kapitel „Allgemeine Client - Projektstruktur [► 591]“ beschrieben ist. Die States: 0,1,11 und 100 sind identisch zu der dort beschriebenen Statemachine. Andere States wurden für das Beispiel modifiziert oder auch neue States hinzugefügt.

Für die Implementierung der Befehle: Anwahl-vor-Ausführung wird ein zusätzlicher, selbstimplementierter Funktionsbaustein benötigt. Im Beispiel wurde dafür ein neuer Funktionsbaustein: „FB_SBOControl“ hinzugefügt. Der Funktionsbaustein: „FB_SBOControl“ wickelt die eigentliche Befehlsausführung für die unterschiedlichen CDC-Typen ab.

Im modifizierten State 10 vom „FB_IEDClient“ sind mehrere IF-Anweisungen aufgeführt, welche über die Instanz des „FB_SBOControl“-Funktionsbausteins die Befehlsausführung des Clients aktivieren. Um sämtliche Zustände des SBO-Befehls (SBO = select before operate) zu berücksichtigen, wird hier ein Enum zur Aktivierung verwendet. Sobald eines der Enums, z. B. „eControl_LLNO_Mod“ nicht „E_AcsiCtlService.None“ ist, wird der Befehl aktiviert. Dies kann der Benutzer über die Online-Werte der Applikation auslösen.

Methoden des "FB_DirectControl"-Funktionsbausteins

| Methodenname | Beschreibung |
|--------------|---|
| APC | Für jeden steuerbaren CDC-Typ ist eine Methode hinzugefügt worden, da die meisten CDCs unterschiedliche Typen verwenden, z. B. „APC“ -> „REAL“ und „SPC“ -> „BOOL“. Weil die CDCs unterschiedliche Typen verwenden, besitzt auch jede dieser Methoden für den Kontrollwert (z. B. „ctlVal“) einen anderen Eingangsparameter. Innerhalb der Methode wird der Wert auf einen Union-Typ geschrieben, welches für jeden CDC-Typ einen Member enthält. Die Union-Variable wird an die „Request“-Methode übergeben. Die „Request“-Methode ist eine private Methode und sie aktiviert die Befehlsausführung. |
| BAC | |
| BSC | |
| DPC | |
| ENC_Mod | |
| INC | |
| ISC | |
| SPC | |
| Execute | |
| Clear | Bricht den Befehl ab und setzt den Funktionsbaustein zurück. |

Der Befehl, abhängig vom CDC-Typen, modifiziert den Kontrollwert des CDCs. Zu Demonstrationszwecken wird z. B. ein Zahlenwert hochgezählt oder ein boolescher Wert getoggelt.

Abhängig von dem Enum-Wert zum Auslösen eines Befehls ändert sich die Ausführung. Das Enum kann folgende Werte annehmen: None, Select, Operate, Select&Operate, Cancel. Bei Select, Operate und Cancel wird nur ein Befehl ausgeführt. Bei Select & Operate wird zuerst ein Select-Befehl durchgeführt, sobald dieser erfolgreich abgeschlossen wurde beginnt der Operate-Befehl. Der Benutzer kann sich für eine dieser Möglichkeiten beim Umschalten des Enums in den Online-Werten entscheiden.

Sobald bei einem der Enum-Werte eine Änderung erfolgte, wird die jeweilige Methode, passend zum CDC-Typ, ausgerufen. Wenn dies erfolgreich beendet wurde, wechselt die Statemachine in den State 11. Dies ist notwendig, da die Bearbeitung der Befehle länger als einen Zyklus in der SPS benötigen. Innerhalb dieses States wird die „Execute“-Methode zyklisch aufgerufen. Anhand des „ipResult“-Schnittstellenzeigers kann ermittelt werden, ob der Befehl noch bearbeitet wird oder bereits abgeschlossen ist. Dies wird über die „IsBusy()“-Methode abgefragt. Sobald der Befehl erfolgreich durchgeführt wurde, wechselt die Statemachine in den State 0.



Dies ist nur eine exemplarische Implementierung der Befehle: Anwahl-vor-Ausführung in der TwinCAT SPS.

7.8 Client - Direct Control with enhanced security

Dieses Beispiel zeigt die exemplarische Implementierung der Direktbefehle-mit-erweiterter-Sicherheit (ctlModel := 3, direct control with enhanced security) in einem TwinCAT IEC 61850 Client-Projekt für folgende Datenklassen der Schaltersteuerung (Common Data Class, CDC):

- SPC (controllable single point) object

- DPC (controllable double point) object
- APC (controllable analogue process value) object
- BAC (binary controlled analogue process value) object
- BSC (binary controlled step position information) object
- ENC (controllable enumerated status) object: Mod
- INC (controllable integer status) object
- ISC (integer controlled step position information) object

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1031/TF6510_TC3_IEC61850/Resources/5494805899.zip

Das hier beschriebene Beispiel nutzt die Statemachine, welche in dem Kapitel „Allgemeine Client - Projektstruktur [▶ 591]“ beschrieben ist. Die States: 0,1,11 und 100 sind identisch zu der dort beschriebenen Statemachine. Andere States wurden für das Beispiel modifiziert oder auch neue States hinzugefügt.

Für die Implementierung der direkten Befehle wird ein zusätzlicher, selbstimplementierter Funktionsbaustein benötigt. Im Beispiel wurde dafür ein neuer Funktionsbaustein: „FB_DIRwControl“ hinzugefügt. Der Funktionsbaustein: „FB_DIRwControl“ wickelt die eigentliche Befehlsausführung für die unterschiedlichen CDC-Typen ab. Im modifizierten State 10 sind mehrere IF-Anweisungen aufgeführt, welche über die Instanz des „FB_DIRwControl“-Funktionsbausteins die Befehlsausführung des Clients aktivieren.

Im Deklarationsteil von „FB_IEDClient“ sind mehrere boolesche Variablen definiert. Eine steigende Flanke an der entsprechenden booleschen Variablen aktiviert in der IF-Anweisung die Befehlsausführung. Dies kann der Benutzer über die Online-Werte der Applikation auslösen.

Methoden des "FB_DIRwControl"-Funktionsbausteins

| Methodenname | Beschreibung |
|--------------|---|
| APC | Für jedes steuerbaren CDC ist eine Methode hinzugefügt worden, da die meisten CDCs unterschiedliche Kontrollwert-Typen verwenden, z.B. „APC“ -> „REAL“ und „SPC“ -> „BOOL“. Weil die CDCs unterschiedliche Typen verwenden, besitzt auch jede dieser Methoden für den Kontrollwert (z. B. „ctlVal“) einen anderen Eingangsparameter. Innerhalb der Methode wird der Wert auf einen Union-Typ geschrieben, welches für jeden CDC einen Member enthält. Die Union-Variable wird an die „Request“-Methode übergeben. Die „Request“-Methode ist eine private Methode und sie aktiviert die Befehlsausführung. |
| BAC | |
| BSC | |
| DPC | |
| ENC_Mod | |
| INC | |
| ISC | |
| SPC | |
| Execute | |
| Clear | Bricht den Befehl ab und setzt den Funktionsbaustein zurück. |

Der Befehl, abhängig vom CDC-Typen, modifiziert den Kontrollwert des CDCs. Zu Demonstrationszwecken wird z. B. ein Zahlenwert hochgezählt oder ein boolescher Wert getoggelt.

Sobald einer der Befehle ausgelöst und erfolgreich abgearbeitet wurde, wird die Statemachine des „FB_IEDClient“-Funktionsbausteins in den State 11 versetzt. Hier verweilt die Statemachine, bis der Befehl erfolgreich durchgeführt wurde. Dafür wird in dem State 11 die „Execute“-Methode in jedem Zyklus aufgerufen und anhand des „ipResult“-Rückgabe-Schnittstellenzeigers kann ermittelt werden, ob der Befehl noch bearbeitet wird oder bereits abgeschlossen ist. Dies wird über die „IsBusy()“-Methode abgefragt. Sobald der Befehl erfolgreich durchgeführt wurde, wechselt die Statemachine in den State 0.

Im Gegensatz zu dem Sample04 mit der beispielhaften Implementierung zu „Direct control with normal security“ erhält der Benutzer beim erfolgreichen Ausführen des „Direct control with enhanced security“ das Event „OnCommandTerminationInd“ am „FB_IEDClient“-Funktionsbaustein. Falls beim Ausführen des Befehls ein Fehler auftritt erhält der Benutzer das Event „OnLastApplErrorInd“.



Dies ist nur eine exemplarische Implementierung der Direktbefehle-mit-erweiterter-Sicherheit in der TwinCAT SPS.

7.9 Client - SBO Control with enhanced security

Dieses Beispiel zeigt die exemplarische Implementierung des Befehls: Anwahl-vor-Ausführung-mit-erweiterter-Sicherheit (ctlModel := 4, select before operate control with enhanced security) in einem TwinCAT IEC 61850 Client-Projekt für folgende Datenklassen der Schaltersteuerung (Common Data Class, CDC):

- SPC (controllable single point) object
- DPC (controllable double point) object
- APC (controllable analogue process value) object
- BAC (binary controlled analogue process value) object
- BSC (binary controlled step position information) object
- ENC (controllable enumerated status) object: Mod
- INC (controllable integer status) object
- ISC (integer controlled step position information) object

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1031/TF6510_TC3_IEC61850/Resources/5494808203.zip

Das hier beschriebene Beispiel nutzt die State Machine, welche in dem Kapitel „Allgemeine Client - Projektstruktur [► 591]“ beschrieben ist. Die States: 0, 1, 11 und 100 sind identisch zu der dort beschriebenen State Machine. Andere States wurden für das Beispiel modifiziert oder auch neue States hinzugefügt.

Für die Implementierung der Befehle: Anwahl-vor-Ausführung-mit-erweiterter-Sicherheit wird ein zusätzlicher, selbstimplementierter Funktionsbaustein benötigt. Im Beispiel wurde dafür ein neuer Funktionsbaustein: „FB_SBOwControl“ hinzugefügt. Der Funktionsbaustein: „FB_SBOwControl“ wickelt die eigentliche Befehlsausführung für die unterschiedlichen CDC-Typen ab.

Im modifizierten State 10 vom „FB_IEDClient“ sind mehrere IF-Anweisungen aufgeführt, welche über die Instanz des „FB_SBOwControl“-Funktionsbausteins die Befehlsausführung des Clients aktivieren. Um sämtliche Zustände des SBO-Befehls (SBO = select before operate) zu berücksichtigen, wird hier ein Enum zur Aktivierung verwendet. Sobald eines der Enums, z. B. „eControl_LLNO_Mod“ nicht „E_AcsiCtlService.None“ ist, wird der Befehl aktiviert. Dies kann der Benutzer über die Online-Werte der Applikation auslösen.

Methoden des "FB_DirectControl"-Funktionsbausteins

| Methodenname | Beschreibung |
|--|---|
| APC
BAC
BSC
DPC
ENC_Mod
INC
ISC
SPC | Für jeden steuerbaren CDC-Typ ist eine Methode hinzugefügt worden, da die meisten CDCs unterschiedliche Typen verwenden, z. B. „APC“ -> „REAL“ und „SPC“ -> „BOOL“. Weil die CDCs unterschiedliche Typen verwenden, besitzt auch jede dieser Methoden für den Kontrollwert (z. B. „ctlVal“) einen anderen Eingangsparameter. Innerhalb der Methode wird der Wert auf einen Union-Typ geschrieben, welches für jeden CDC einen Member enthält. Die Union-Variable wird an die „Request“-Methode übergeben. Die „Request“-Methode ist eine private Methode und sie aktiviert die Befehlsausführung. |
| Execute | Sobald die „Request“-Methode des Funktionsbausteins erfolgreich aufgerufen wurde, muss die „Execute“-Methode in jeden Zyklus aufgerufen werden. Innerhalb dieser Methode befindet sich eine separate Statemachine, welche unabhängig von der Statemachine im Client-Funktionsbaustein arbeitet. Hierdurch wird der Code in kleinere Abschnitte aufgeteilt und kann in Zukunft besser erweitert werden. |
| Clear | Bricht den Befehl ab und setzt den Funktionsbaustein zurück. |

Der Befehl, abhängig vom CDC-Typen, modifiziert den Kontrollwert des CDC. Zu Demonstrationszwecken wird z. B. ein Zahlenwert hochgezählt oder ein boolescher Wert getoggelt.

Abhängig von dem Enum-Wert zum Auslösen eines Befehls ändert sich die Ausführung. Das Enum kann folgende Werte annehmen: None, Select, Operate, Select&Operate, Cancel. Bei Select, Operate und Cancel wird nur ein Befehl ausgeführt. Bei Select & Operate wird zuerst ein Select-Befehl durchgeführt, sobald dieser erfolgreich abgeschlossen wurde beginnt der Operate-Befehl. Der Benutzer kann sich für eine dieser Möglichkeiten beim Umschalten des Enums in den Online-Werten entscheiden.

Sobald bei einem der Enum-Werte eine Änderung erfolgte, wird die jeweilige Methode, passend zum CDC-Typ, ausgerufen. Wenn dies erfolgreich beendet wurde, wechselt die Statemachine in den State 11. Dies ist notwendig, da die Bearbeitung der Befehle länger als einen Zyklus in der SPS benötigen. Innerhalb dieses States wird die „Execute“-Methode zyklisch aufgerufen. Anhand des „ipResult“-Schnittstellenzeigers kann ermittelt werden, ob der Befehl noch bearbeitet wird oder bereits abgeschlossen ist. Dies wird über die „IsBusy()“-Methode abgefragt. Sobald der Befehl erfolgreich durchgeführt wurde, wechselt die Statemachine in den State 0.

Im Gegensatz zu dem Sample05 mit der beispielhaften Implementierung zu „Select before operate control with normal security“ erhält der Benutzer beim erfolgreichen Ausführen des „Select before operate with enhanced security“ das Event „OnCommandTerminationInd“ am „FB_IEDClient“-Funktionsbaustein. Falls beim Ausführen des Befehls ein Fehler auftritt erhält der Benutzer das Event „OnLastApplErrorInd“.



Dies ist nur eine exemplarische Implementierung der Befehle: Anwahl-vor-Ausführung-mit-erweiterter-Sicherheit in der TwinCAT SPS.

7.10 Client - Read/Write DataSet Values (GetDataSetValues, SetDataSetValues)

Dieses Beispiel zeigt die Verwendung der „GetDataSetValuesReq“- und „SetDataSetValuesReq“- Methoden des Client-Funktionsbausteins.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1031/TF6510_TC3_IEC61850/Resources/8922514571.zip

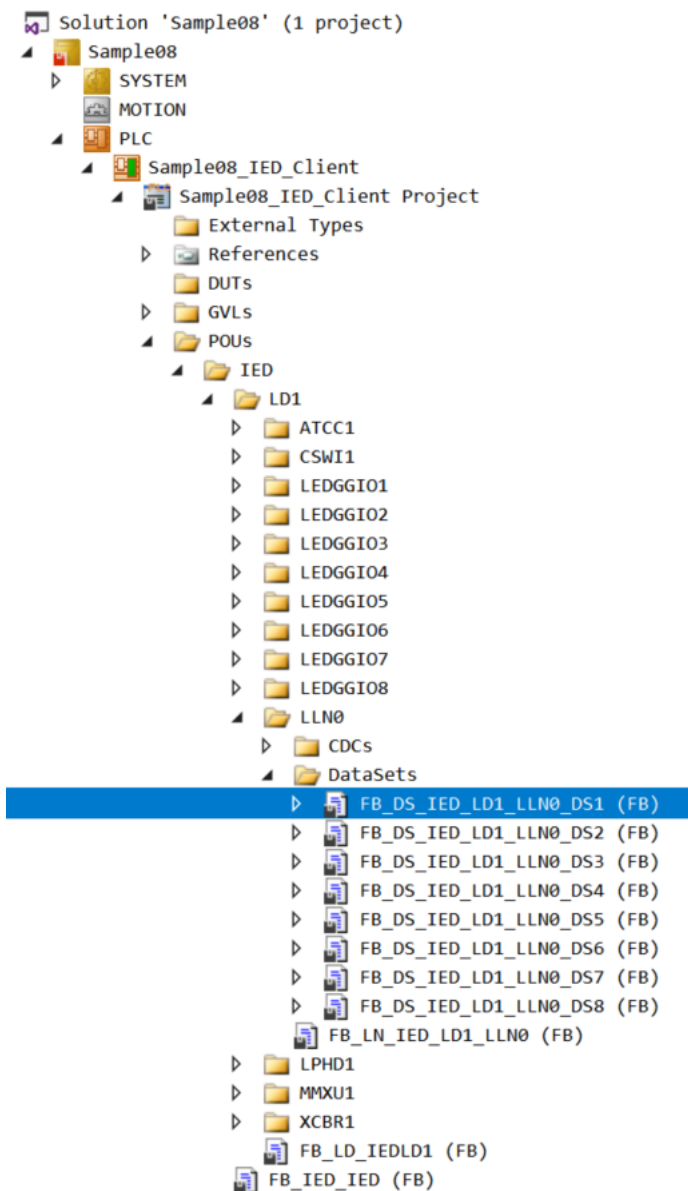
Das hier beschriebene Beispiel nutzt die Statemachine, welche in dem Kapitel „Allgemeine Client - Projektstruktur [► 591]“ beschrieben ist. Die States: 0,1,11 und 100 sind identisch zu der dort beschriebenen Statemachine. Andere States wurden für das Beispiel modifiziert oder auch neue States hinzugefügt.

Allgemeine Information über IEC 61850 DataSets

Ein DataSet ist eine Liste von Datenattributen oder Daten die gemeinsam (meistens in einem Report- oder einer GOOSE-Meldung) übertragen werden können. Die einzelnen Listeneinträge eines DataSets werden DataSet-Member genannt. Die DataSet-Member werden im TwinCAT Telecontrol Configurator konfiguriert. Der SPS Code der DataSets und DataSet-Member wird vom Konfigurator automatisch während der TwinCAT Projektgenerierung erzeugt. Mit einem DataSet können mehrere Daten oder Datenattribute gleichzeitig gelesen oder geschrieben werden. Die Client-Methode „GetDataSetValuesReq“ kann alle im DataSet definierten DataSet-Member gemeinsam auslesen (die Werte werden vom Server zum Client übertragen). Mit der Client-Methode „SetDataSetValuesReq“ können alle im DataSet definierten DataSet-Member beschrieben werden (die Werte werden vom Client zum Server übertragen).

Vorzugsweise befinden sich die DataSets im IEC 61850 Datenmodell unterhalb des logischen Knoten LLN0. Ein DataSet kann theoretisch aber auch jedem anderen logischen Knoten zugeordnet (mit diesem Knoten verlinkt) werden.

Folgende Abbildung zeigt die, während der Projektgenerierung erstellten DataSet-Funktionsbausteine im TwinCAT Projektbaum:



Die abgebildeten DataSet-Funktionsbausteine gehören zum LLN0 und aus diesem Grund werden sie auch im Deklarationsteil vom LLN0 instanziiert. Die Instanzen der DataSets: „DS1“ bis „DS8“ und die Verknüpfung mit dem logischen Knoten wurde ebenfalls automatisch generiert.


```

1 FUNCTION_BLOCK FB_LN_IED_LD1_LLN0 EXTENDS FB_AcsiCommonLogicalNodeClass
2 VAR_INPUT
3   → NamPlt: FB_DO_IED_LD1_LLN0_NamPlt := (sClass:='LPL', bLinkResult:=THIS^.AddDataToContainer(ipData:=NamPlt));
4   → Beh: FB_DO_IED_LD1_LLN0_Beh := (sClass:='ENS', bLinkResult:=THIS^.AddDataToContainer(ipData:=Beh));
5   → Health: FB_DO_IED_LD1_LLN0_Health := (sClass:='ENS', bLinkResult:=THIS^.AddDataToContainer(ipData:=Health));
6   → Mod_: FB_DO_IED_LD1_LLN0_Mod := (sObjectName:='Mod', sClass:='ENC', bLinkResult:=THIS^.AddDataToContainer(ipData:=Mod_));
7   →
8   → DS1: FB_DS_IED_LD1_LLN0_DS1 := (bLinkResult:=THIS^.AddDataSetToContainer(ipDataSet:=DS1));
9   → DS2: FB_DS_IED_LD1_LLN0_DS2 := (bLinkResult:=THIS^.AddDataSetToContainer(ipDataSet:=DS2));
10  → DS3: FB_DS_IED_LD1_LLN0_DS3 := (bLinkResult:=THIS^.AddDataSetToContainer(ipDataSet:=DS3));
11  → DS4: FB_DS_IED_LD1_LLN0_DS4 := (bLinkResult:=THIS^.AddDataSetToContainer(ipDataSet:=DS4));
12  → DS5: FB_DS_IED_LD1_LLN0_DS5 := (bLinkResult:=THIS^.AddDataSetToContainer(ipDataSet:=DS5));
13  → DS6: FB_DS_IED_LD1_LLN0_DS6 := (bLinkResult:=THIS^.AddDataSetToContainer(ipDataSet:=DS6));
14  → DS7: FB_DS_IED_LD1_LLN0_DS7 := (bLinkResult:=THIS^.AddDataSetToContainer(ipDataSet:=DS7));
15  → DS8: FB_DS_IED_LD1_LLN0_DS8 := (bLinkResult:=THIS^.AddDataSetToContainer(ipDataSet:=DS8));
16  →
17  → urcb101: FB_ScsmUrCbImplClass := (RptID:=(sValue:='IEDLD1/LLN0.urcb101'), DatSet:=(sValue:='IEDLD1/LLN0.DS1'), ConfRev:=
18  → brcb101: FB_ScsmBrCbImplClass := (RptID:=(sValue:='IEDLD1/LLN0.brcb101'), DatSet:=(sValue:='IEDLD1/LLN0.DS1'), ConfRev:=
19 END_VAR
20 VAR
21 END_VAR
    
```

Der SPS-Code für die Konfiguration der DataSet-Member wird wie die DataSets selbst ebenfalls automatisch generiert. Dieser Code befindet sich in der OnInit-Methode des jeweiligen DataSet-Funktionsbausteins (siehe Bild unten).

The screenshot shows the Solution Explorer on the left with a tree view of the project structure. Under 'DataSets', several DataSet objects are listed, including 'FB_DS_IED_LD1_LLN0_DS8'. The right pane shows the 'OnInit' method code for 'FB_DS_IED_LD1_LLN0_DS8', which is a Boolean function. The code includes a loop of 'IF NOT AddMember' statements for each DataSet member (e.g., LEDGG106, LEDGG107, LEDGG108, CDCs, and various DS1-DS8) to ensure they are properly initialized and added to the container.

Statische DataSets vs. dynamische DataSets

Die TwinCAT Projektgenerierung erzeugt statische, nicht löschbare DataSets (static created persistent data-sets). D.h. diese DataSets sind im Datenmodell immer verfügbar und können und sollen nicht gelöscht werden. Es gibt aber auch dynamische DataSets, die während der Kommunikation erstellt werden können. Die dynamisch erstellten DataSets können wieder gelöscht werden. Sie werden gelöscht entweder automatisch, beim Verbindungsabbruch (dynamic created non-persistent data-sets) oder zum späteren Zeitpunkt, auch von einem anderen Client (dynamic created persistent data-sets).

Beispielprojekt

In dem zip-Archiv befindet sich eine ICD-Konfigurationsdatei. Diese Datei kann verwendet werden, um z. B. einen Server mit Hilfe einer Drittherstellersoftware zu simulieren oder mit TwinCAT Telecontrol Configurator ein neues oder modifiziertes TwinCAT Projekt zu erzeugen.

Falls Sie vorhaben, ein DataSet zu verwenden, um mehrere Daten oder Datenattribute gleichzeitig zu beschreiben dann müssen Sie noch Folgendes beachten: Alle konfigurierten DataSet-Member müssen auch einen Schreibzugriff erlauben, andernfalls wird der Schreibzugriff fehlschlagen. Dies hängt hauptsächlich von der funktionalen Gruppe (FC) des DataSet-Members ab. In diesem Beispiel wurde zur Demonstration für den Schreibzugriff das DataSet „DS8“ so konfiguriert, dass alle Dataset-Member beschrieben werden können (funktionale Gruppe: „DC“). Bei unserem Client-Beispielprojekt muss natürlich der Schreibzugriff auf alle DataSet-Member auf der Serverseite möglich sein.

Im Funktionsbaustein FB_IEDClient sind mehrere booleschen Variablen definiert. Eine steigende Flanke an einer dieser Variablen aktiviert die Methode: „GetDataSetValuesReq“ oder „DataSetValuesReq“ mit dem jeweiligen DataSet als Eingangsparameter. Als Beispiel: Eine steigende Flanke an der Variablen „bGetDataSetValues_LLNO_DS1“ aktiviert den Befehl zum Lesen des DataSet: „DS1“ und eine steigende Flanke an „bDataSetValues_LLNO_DS8“ aktiviert den Befehl zum Schreiben des DataSet: „DS8“.

```
...
bGetDataSetValues_LLNO_DS1 : BOOL := TRUE;
bGetDataSetValues_LLNO_DS2 : BOOL := TRUE;
bGetDataSetValues_LLNO_DS3 : BOOL := TRUE;
bGetDataSetValues_LLNO_DS4 : BOOL := TRUE;
bGetDataSetValues_LLNO_DS5 : BOOL := TRUE;
bGetDataSetValues_LLNO_DS6 : BOOL := TRUE;
bGetDataSetValues_LLNO_DS7 : BOOL := TRUE;
bGetDataSetValues_LLNO_DS8 : BOOL := TRUE;
bDataSetValues_LLNO_DS8 : BOOL := TRUE;
...
```

Es folgt ein Auszug aus dem Beispielcode zum Lesen des DataSet: „DS8“. Zuerst aktiviert der Methodenaufruf „GetDataSetValuesReq“ die Übertragung der Daten vom Server zum Client. Danach, in weiteren SPS-Zyklen muss die Methode „ipResult.Execute()“ so lange aufgerufen werden, bis die Übertragung abgeschlossen wurde. „ipResult.IsBusy()“ liefert in diesem Fall FALSE. Die empfangenen Datenwerte oder Attributwerte der DataSet-Member werden beim Erfolg automatisch ins Client-Datenmodell kopiert (gemappt).

```
...
ELSIF bGetDataSetValues_LLNO_DS8 THEN
    bGetDataSetValues_LLNO_DS8:= FALSE;
    bSuccess:= fbConnection.GetDataSetValuesReq(ipDataSet:=fbIED.IEDLD1.LLNO.DS8, hUser:=0, ipSink:=
0, nInvokeID=>nInvokeID, ipResult=>ipResult);
    state:= SEL(bSuccess, 100, 11);
...
```

Im Beispielcode zum Schreiben des DataSet: „DS8“ werden zum Demonstrationszweck jedes Mal die zu schreibenden Daten des DataSet: „DS8“ modifiziert. Zuerst werden neue Datenwerte oder Attributwerte den DataSet-Membern im Client-Datenmodell zugewiesen. Danach aktiviert der Methodenaufruf „DataSetValuesReq“ die Übertragung der Daten zum Server. In weiteren SPS-Zyklen muss analog wie schon oben beschrieben die Methode „ipResult.Execute()“ so lange aufgerufen werden, bis die Übertragung abgeschlossen wurde. Die neuen Datenwerte oder Attributwerte werden auf der Serverseite ins Server-Datenmodell kopiert (gemappt).

```
...
ELSIF bDataSetValues_LLNO_DS8 THEN
    bDataSetValues_LLNO_DS8:= FALSE;
    (* As example we modify some description data values *)
    sConfigRev:= TO_STRING(nRev:=nRev+1);
    sHwRev:= TO_STRING(TO_REAL(nRev));
    sSwRev:= TO_STRING(TO_REAL(nRev));

    fbIED.IEDLD1.LPHD1.PhyNam.vendor.sValue:= sVendor;
    fbIED.IEDLD1.LPHD1.PhyNam.hwRev.sValue:= sHwRev;
    fbIED.IEDLD1.LPHD1.PhyNam.swRev.sValue:= sSwRev;
    fbIED.IEDLD1.LPHD1.PhyNam.serNum.sValue:= sSerNum;
    fbIED.IEDLD1.LPHD1.PhyNam.model.sValue:= sModel;
    fbIED.IEDLD1.LPHD1.PhyNam.location.sValue:= sLocation;

    fbIED.IEDLD1.LLNO.NamPlt.vendor.sValue:= sVendor;
    fbIED.IEDLD1.LLNO.NamPlt.swRev.sValue:= sSwRev;
    fbIED.IEDLD1.LLNO.NamPlt.d.sValue:= 'LLNO demo node';
    fbIED.IEDLD1.LLNO.NamPlt.configRev.sValue:= sConfigRev;

    fbIED.IEDLD1.MMXU1.NamPlt.vendor.sValue:= sVendor;
    fbIED.IEDLD1.MMXU1.NamPlt.swRev.sValue:= sSwRev;
    fbIED.IEDLD1.MMXU1.NamPlt.d.sValue:= 'MMXU1 demo node';
    fbIED.IEDLD1.MMXU1.NamPlt.configRev.sValue:= sConfigRev;

    fbIED.IEDLD1.XCBR1.NamPlt.vendor.sValue:= sVendor;
    fbIED.IEDLD1.XCBR1.NamPlt.swRev.sValue:= sSwRev;
    fbIED.IEDLD1.XCBR1.NamPlt.d.sValue:= 'XCBR1 demo node';
    fbIED.IEDLD1.XCBR1.NamPlt.configRev.sValue:= sConfigRev;

    fbIED.IEDLD1.CSWI1.NamPlt.vendor.sValue:= sVendor;
    fbIED.IEDLD1.CSWI1.NamPlt.swRev.sValue:= sSwRev;
    fbIED.IEDLD1.CSWI1.NamPlt.d.sValue:= 'CSWI1 demo node';
    fbIED.IEDLD1.CSWI1.NamPlt.configRev.sValue:= sConfigRev;
...
```



```

fbIED.IEDLD1.LEDGGIO1.NamPlt.vendor.sValue:= sVendor;
fbIED.IEDLD1.LEDGGIO1.NamPlt.swRev.sValue:= sSwRev;
fbIED.IEDLD1.LEDGGIO1.NamPlt.d.sValue:= 'LEDGGIO1 demo node';

fbIED.IEDLD1.LEDGGIO2.NamPlt.vendor.sValue:= sVendor;
fbIED.IEDLD1.LEDGGIO2.NamPlt.swRev.sValue:= sSwRev;
fbIED.IEDLD1.LEDGGIO2.NamPlt.d.sValue:= 'LEDGGIO2 demo node';

fbIED.IEDLD1.LEDGGIO3.NamPlt.vendor.sValue:= sVendor;
fbIED.IEDLD1.LEDGGIO3.NamPlt.swRev.sValue:= sSwRev;
fbIED.IEDLD1.LEDGGIO3.NamPlt.d.sValue:= 'LEDGGIO3 demo node';

fbIED.IEDLD1.LEDGGIO4.NamPlt.vendor.sValue:= sVendor;
fbIED.IEDLD1.LEDGGIO4.NamPlt.swRev.sValue:= sSwRev;
fbIED.IEDLD1.LEDGGIO4.NamPlt.d.sValue:= 'LEDGGIO4 demo node';

fbIED.IEDLD1.LEDGGIO5.NamPlt.vendor.sValue:= sVendor;
fbIED.IEDLD1.LEDGGIO5.NamPlt.swRev.sValue:= sSwRev;
fbIED.IEDLD1.LEDGGIO5.NamPlt.d.sValue:= 'LEDGGIO5 demo node';

fbIED.IEDLD1.LEDGGIO6.NamPlt.vendor.sValue:= sVendor;
fbIED.IEDLD1.LEDGGIO6.NamPlt.swRev.sValue:= sSwRev;
fbIED.IEDLD1.LEDGGIO6.NamPlt.d.sValue:= 'LEDGGIO6 demo node';

fbIED.IEDLD1.LEDGGIO7.NamPlt.vendor.sValue:= sVendor;
fbIED.IEDLD1.LEDGGIO7.NamPlt.swRev.sValue:= sSwRev;
fbIED.IEDLD1.LEDGGIO7.NamPlt.d.sValue:= 'LEDGGIO7 demo node';

fbIED.IEDLD1.LEDGGIO8.NamPlt.vendor.sValue:= sVendor;
fbIED.IEDLD1.LEDGGIO8.NamPlt.swRev.sValue:= sSwRev;
fbIED.IEDLD1.LEDGGIO8.NamPlt.d.sValue:= 'LEDGGIO8 demo node';

fbIED.IEDLD1.ATCC1.NamPlt.vendor.sValue:= sVendor;
fbIED.IEDLD1.ATCC1.NamPlt.swRev.sValue:= sSwRev;
fbIED.IEDLD1.ATCC1.NamPlt.d.sValue:= 'ATCC1 demo node';
fbIED.IEDLD1.ATCC1.NamPlt.configRev.sValue:= sConfigRev;

bSuccess:= fbConnection.SetDataSetValuesReq(ipDataSet:=fbIED.IEDLD1.LLN0.DS8, hUser:=0, ipSink:=
0, nInvokeID=>nInvokeID, ipResult=>ipResult);
state:= SEL(bSuccess, 100, 11);
...
...
CASE state OF
...
11:
    IF ipResult <> 0 THEN
        ipResult.Execute();
        IF NOT (bBusy:=ipResult.IsBusy()) THEN
            state:= SEL(ipResult.IsCompleted(), 100(* failed or aborted *), 0(* succeeded *));
        END_IF
    END_IF
...
END_CASE
...

```

7.11 Client - GOOSE Subscriber (GetGoCBDataValues, SetGoCBDataValues)

Dieses Beispiel zeigt, wie ein GOOSE-Steuerblock auf der Serverseite mit Hilfe der Dienste GetGoCBDataValues und SetGoCBDataValues kontrolliert wird. Die Dienste ermöglichen das Lesen und/oder das Schreiben der Attributwerte des GOOSE-Steuerblocks. Beim Setzen des Attributes „GoEna“ auf „TRUE“ wird das Versenden (publishing) der GOOSE-Nachrichten auf der Serverseite aktiviert. Beim Setzen des Attributes „GoEna“ auf „FALSE“ wird das Versenden der GOOSE-Nachrichten deaktiviert.

Ein Subscriber, der die GOOSE-Nachrichten auf der Client-Seite empfängt, ist im Beispiel ebenfalls enthalten.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1031/TF6510_TC3_IEC61850/Resources/7280990219.zip

Das hier beschriebene Beispiel nutzt die State Machine, welche in dem Kapitel „Allgemeine Client - Projektstruktur [▶ 591]“ beschrieben ist. Die States: 0,1,11 und 100 sind identisch zu der dort beschriebenen State Machine. Andere States wurden für das Beispiel modifiziert oder auch neue States hinzugefügt.

Im Projektbaum unter dem Zweig I/O-Device finden Sie eine Netzwerkadapterinstanz mit dem Namen „GSE (RT-Ethernet Adapter)“. Diese Adapterinstanz muss entsprechend konfiguriert werden, d.h. die I/O-Konfiguration muss an die vorhandene Hardware und auf die Zielplattform, auf der das Projekt laufen soll, angepasst werden.

Eine erneute I/O-Konfiguration ist auch dann nötig, wenn Sie die Zielplattform wechseln. Diese Konfiguration muss manuell in TwinCAT XAE vorgenommen werden. Zusätzlich zu der I/O-Konfiguration des Netzwerkadapters muss ein Link zwischen dem Netzwerkadapter und den SPS-Bausteinen für die Goose-Kommunikation hergestellt werden. Mit dem Link können die vom Netzwerkadapter empfangenen Daten an die Instanz des Funktionsbausteins: „FB_[IEDName]Gse“ weitergeleitet werden. In umgekehrter Richtung kann die Instanz des Funktionsbausteins „FB_[IEDName]Gse“ die zu sendenden Daten an den Netzwerkadapter weiterleiten.

Hier finden Sie weitere Informationen: [RT – Ethernet Adapter Konfiguration](#) [► 37].

7.12 Client - GOOSE Subscriber (ohne Client-Server-Kommunikation)

In einem Client Projekt wird von dem TwinCAT Telecontrol Configurator standardmäßig auch ein Client-Baustein generiert (siehe: „Allgemeine Client -Projektstruktur“ [► 591]). Der Client-Baustein kann dann z.B. dazu verwendet werden, um den GOOSE-Publisher auf der Serverseite via Client-Server-Dienste wie „GetGoCBDataValues“ oder „SetGoCBDataValues“ zu aktivieren/deaktivieren oder die GoCBs zu konfigurieren. In manchen Fällen soll aber auf eine Client-Server-Kommunikation mit dem Gerät verzichtet werden und nur ein „reiner“ GOOSE Subscriber implementiert werden. Ein solcher Subscriber kann nach dem SPS-Programmstart automatisch mit dem Empfang der GOOSE-Nachrichten beginnen. Dieses Beispiel zeigt die Implementierung eines Subscribers, aber ohne den Client-Server-Kommunikationsteil. Die nicht benötigten Teile des Client-Projekts wurden in diesem Beispiel bewusst entfernt.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1031/TF6510_TC3_IEC61850/Resources/7282249995.zip

In der globalen Variablenliste wird eine IED-Datenmodell-Bausteininstanz („fbIED“) und eine oder mehrere GSE-Bausteininstanzen für GOOSE-Kommunikation und GSE-Management („fbIEDGse“) benötigt. Dieses Beispiel verwendet nur einen Netzwerkadapter für die GOOSE-Kommunikation und drei GoCBs im IED-Datenmodell.

Namensraum: TcTelecontrol

Typ: Globale Variablenliste (GVL)

```
VAR_GLOBAL
  ipCreator : I_AcsiCodeCreatorClass := GVL_AcsiVars.Creator.SetCodeRev(codeRev:=2).SetGuiVer(major:=1, minor:=0, build:=93, revision:=10);
  fbIED     : FB_IED_IED;
  fbIEDGse  : FB_IEDGse := (fbAdapter:=(ipIED:=fbIED, settings:=(sMulticastAddr:='01-0C-CD-01-00-01', eDispatchMode:=E_GseDispatchMode.NonPromiscuous));
END_VAR
```

Das „MAIN“-Programm wird zyklisch von einer TwinCAT Task aufgerufen und muss nur das Programm „P_IEC61850MAIN“ aufrufen. „P_IEC61850MAIN“ ruft wiederum den GSE-Baustein auf, der für den Empfang der GOOSE-Nachrichten und das Mapping der empfangenen GOOSE-Daten ins IED-Datenmodell zuständig ist.

```
PROGRAM MAIN
VAR
END_VAR

P_IEC61850MAIN();

PROGRAM P_IEC61850MAIN
VAR
END_VAR

fbIEDGse();
```

In der Beispielimplementierung des GSE-Bausteins wird der Subscriber-Prozess bei allen drei GoCBs nach dem SPS-Start automatisch aktiviert. Eine steigende Flanke an der „bUnsubscribe“-Variablen deaktiviert den Subscriber-Prozess der GoCBs.

```

FUNCTION_BLOCK FB_IEDGse IMPLEMENTS I_GseLinkStatusEventSink
VAR_INPUT
    fbAdapter      : FB_GseAdapterClass := (ipLinkStatus:=THIS^);
END_VAR
VAR
    eLinkStatus   : E_GseLinkStatus;
    bSuccess      : BOOL;
    ipError       : I_ServiceErrorClass;
    bSubscribe    : BOOL := TRUE;
    bUnsubscribe  : BOOL;
END_VAR

bSuccess:= fbAdapter.Execute(ipError=>ipError);

IF bSubscribe THEN
    bSubscribe:= FALSE;
    bSuccess:= fbIED.IEDLD1.LLN0.gocb01.Subscriber.Enable(ipAdapter:=fbAdapter, ipError=>ipError);
    bSuccess:= fbIED.IEDLD1.LLN0.gocb02.Subscriber.Enable(ipAdapter:=fbAdapter, ipError=>ipError);
    bSuccess:= fbIED.IEDLD1.LLN0.gocb03.Subscriber.Enable(ipAdapter:=fbAdapter, ipError=>ipError);
ELSIF bUnsubscribe THEN
    bUnsubscribe:= FALSE;
    bSuccess:= fbIED.IEDLD1.LLN0.gocb01.Subscriber.Disable(ipError=>ipError);
    bSuccess:= fbIED.IEDLD1.LLN0.gocb02.Subscriber.Disable(ipError=>ipError);
    bSuccess:= fbIED.IEDLD1.LLN0.gocb03.Subscriber.Disable(ipError=>ipError);
ELSE
    bSuccess:= fbIED.IEDLD1.LLN0.gocb01.Subscriber.Execute(ipError=>ipError);
    bSuccess:= fbIED.IEDLD1.LLN0.gocb02.Subscriber.Execute(ipError=>ipError);
    bSuccess:= fbIED.IEDLD1.LLN0.gocb03.Subscriber.Execute(ipError=>ipError);
END_IF

```

Der GSE-Baustein implementiert die „I_GseLinkStatusEventSink“-Schnittstelle. Die Methode: „OnLinkStatusChange“ gehört zu dieser Schnittstellenimplementierung und wird immer dann aufgerufen, wenn sich der Status der Netzwerkverbindung (am Netzwerkadapter) ändert. Die SPS-Applikation kann z. B. den Netzwerkverbindungsstatus via „eLinkStatus“-Variable abfragen oder überprüfen.

```

METHOD OnLinkStatusChange
VAR_INPUT
    ipAdapter      : I_GseAdapterClass;
    eStatus       : E_GseLinkStatus;
END_VAR
VAR
END_VAR
END_VAR

eLinkStatus:= eStatus;

```

Im Projektbaum unter dem Zweig I/O-Device finden Sie eine Netzwerkadapterinstanz mit dem Namen „GSE (RT-Ethernet Adapter)“. Diese Adapterinstanz muss entsprechend konfiguriert werden, d.h. die I/O-Konfiguration muss an die vorhandene Hardware und auf die Zielplattform, auf der das Projekt laufen soll, angepasst werden.

Eine erneute I/O-Konfiguration ist auch dann nötig, wenn Sie die Zielplattform wechseln. Diese Konfiguration muss manuell in TwinCAT XAE vorgenommen werden. Zusätzlich zu der I/O-Konfiguration des Netzwerkadapters muss ein Link zwischen dem Netzwerkadapter und den SPS-Bausteinen für die Goose-Kommunikation hergestellt werden. Mit dem Link können die vom Netzwerkadapter empfangenen Daten an die Instanz des Funktionsbausteins: „FB_[IEDName]Gse“ weitergeleitet werden. In umgekehrter Richtung kann die Instanz des Funktionsbausteins „FB_[IEDName]Gse“ die zu sendenden Daten an den Netzwerkadapter weiterleiten.

Hier finden Sie weitere Informationen: [RT – Ethernet Adapter Konfiguration \[► 37\]](#).

7.13 Client - User defined timestamp clock source

In einigen Applikationen soll z.B. neben dem Sollwert auch ein Zeitstempel verwendet werden, der von einer externen Quelle stammt (z.B. einer GPS-Uhr). Dieses Beispiel zeigt die Implementierung einer Benutzerdefinierten Uhr/Uhrzeit für Zeitstempelungsaufgaben.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1031/TF6510_TC3_IEC61850/Resources/10698182027.zip

Das hier beschriebene Beispiel nutzt die State Machine, welche in dem Kapitel „[Allgemeine Client - Projektstruktur \[► 591\]](#)“ beschrieben ist. Die States: 0,1,11 und 100 sind identisch zu der dort beschriebenen State Machine. Andere States wurden für das Beispiel modifiziert oder auch neue States hinzugefügt.

Die TwinCAT IEC 61850 Client-Implementierung verwendet die Schnittstelle:

„I_ScsmSystemClockEventSink“, um die Uhrzeit für Zeitstempelungsaufgaben abzufragen. Es ist möglich einen eigenen Funktionsbaustein zu verwenden, der diese spezielle Schnittstelle implementiert und den Client-Baustein so konfiguriert, dass dieser die neue Zeitquelle verwendet. In der globalen Variablenliste „TcTelecontrol“ wird dem Client-Baustein durch das Setzen der Eigenschaft „ipSystemClock:=fbMyClock“ die neue Zeitquelle zugewiesen. Der Client-Baustein wird danach jedes Mal die Methode: „OnGetSystemTime“ aufrufen, wenn von ihm ein neuer Zeitstempel benötigt wird. Die IEC 61850 Client-Server-Kommunikation verwendet Zeitstempel in mindestens zwei verschiedenen Formaten: UTC-Time und Binary-Time. Aus diesem Grund liefert diese Methode den Zeitstempel als Ausgangsvariablen in diesen zwei Formaten.

```
VAR_GLOBAL
  ipCreator : I_AcsiCodeCreatorClass := GVL_AcsiVars.Creator.SetCodeRev(codeRev:=2).SetGuiVer(major:=1, minor:=0, build:=93, revision:=10);
  fbIED : FB_IED_IED;
  (* User defined clock (time stamp source) *)
  fbMyClock : FB_MyClock;
  fbIEDClient : FB_IEDClient := (fbConnection:=(ipIED:=fbIED, ipSystemClock:=fbMyClock, settings:=(sRemoteHost:='127.0.0.1')));
END_VAR
```

Das Beispiel demonstriert eine sehr einfache Software-Uhr, die mit Hilfe des RTC_EX-Funktionsbausteins realisiert wurde.

```
FUNCTION_BLOCK FB_MyClock IMPLEMENTS I_ScsmSystemClockEventSink
VAR
  _tT : T_UtcTime:=(secondSinceEpoch:=DT#2021-04-01-00:00:00, quality:=(ClockNotSynchronized:=TRUE, ClockFailure:=FALSE, LeapSecondsKnown:=FALSE), fractionOfSecond:=[0,0,0]);(* Actual UTC time. *)
  _tB : T_BinaryTime;(* Actual binary-time (EntryTime) *)
  clock : RTC_EX:=(EN:=TRUE, PDT:=DT#2021-04-01-00:00:00, PMSEK:=0);
  refreshTimer : TON:=(IN:=TRUE, PT:=T#1S);
END_VAR

METHOD FINAL Execute : BOOL
VAR_INPUT
END_VAR

refreshTimer();
IF Execute:=refreshTimer.Q THEN
  refreshTimer(IN:=FALSE); refreshTimer(IN:=TRUE);
  Update();
END_IF

METHOD FINAL OnGetSystemTime : BOOL
VAR_INPUT
  ipAA: I_ScsmAssociationClass;(* Application association. If = 0 => optional or unknown. *)
END_VAR
VAR_OUTPUT
  tT: T_UtcTime;(* UTC-time. *)
  tB: T_BinaryTime;(* Binary-time *)
END_VAR

tT:=_tT;
tB:=_tB;
OnGetSystemTime:=TRUE;

METHOD FINAL SetClock
VAR_INPUT
  tSet : DT;(* New time to set *)
END_VAR

clock(EN:=FALSE);
clock(EN:=TRUE, PDT:=tSet);
_tT.quality.ClockNotSynchronized:=FALSE;
Update();

METHOD FINAL Update
VAR_INPUT
END_VAR

clock();(* update clock time *)

(* convert to utc-time format *)
_tT.secondSinceEpoch:=clock.CDT;
_tT.fractionOfSecond:=LTIME_TO_UtcTimeFractionOfSecond(in:=TIME_TO_LTIME(DWORD_TO_TIME(clock.CMSEK)));
Accuracy_To_UtcTimeQualityAccuracy(in:=E_UtcTimeAccuracy_03, bAccuracy0=>_tT.quality.Accuracy0, bAccuracy1=>_tT.quality.Accuracy1, bAccuracy2=>_tT.quality.Accuracy2, bAccuracy3=>_tT.quality.Accuracy3, bAccuracy4=>_tT.quality.Accuracy4);

(* convert to binary-time format *)
_tB.day:=Date_To_BinaryTime6Day(in:=DT_TO_DATE(clock.CDT));
_tB.timeOfDay:=DT_TO_TOD(clock.CDT) + DWORD_TO_TIME(clock.CMSEK);
```

7.14 Client - Dynamic created DataSets (CreateDataSet, DeleteDataSet)

Dieses Beispiel zeigt die Verwendung der „CreateDataSetReq“- und „DeleteDataSet“-Methoden des Client-Funktionsbausteins.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1031/TF6510_TC3_IEC61850/Resources/11311587339.zip

Das hier beschriebene Beispiel nutzt die StateMachine, welche in dem Kapitel „Allgemeine Client - Projektstruktur |> 591|“ beschrieben ist. Die States: 0, 1, 11 und 100 sind identisch zu der dort beschriebenen StateMachine. Andere States wurden für das Beispiel modifiziert oder auch neue States hinzugefügt.

Allgemeine Information über die dynamischen DataSets

Die Instanzen der statischen, während der Laufzeit immer verfügbaren DataSets, werden in TwinCAT Telecontrol Configurator entweder vom Anwender konfiguriert oder aus einer ICD-Datei importiert. Die SPS-Projektgenerierung erstellt später den nötigen SPS-Code für die Instanziierung dieser DataSets. Dieser Code beinhaltet auch die Konfiguration der DataSet-Member und die Zuordnung des DataSets zu einem logischen Knoten im IEC 61850 Datenmodell. Die Konfiguration der statischen DataSets darf zur Laufzeit des SPS-Programms nicht verändert und die DataSets können auch nicht gelöscht werden.

Manchmal soll ein DataSet dynamisch erstellt und nur für eine kurze Zeit (z.B. solange die Kommunikationsverbindung steht) genutzt werden. In dieser Zeit ist es möglich, einige Messwerte mit Hilfe eines Reports zu überwachen. Wenn die Verbindung nicht mehr vorhanden ist soll dieser DataSet wieder aus dem Datenmodell entfernt/gelöscht werden.

Mit Hilfe der Client-Methoden: „CreateDataSetReq“ und „DeleteDataSetReq“ können DataSets dynamisch (zur Laufzeit) erstellt, konfiguriert und wieder gelöscht werden. Die dynamisch erstellten DataSets können entweder als „persistent“ oder „non-persistent“ DataSets konfiguriert werden. Die „non-persistent“ DataSets sind einer Client-Server-Verbindung fest zugeordnet und existieren nur maximal so lange, bis die Verbindung unterbrochen wurde. Die „non-persistent“ DataSets einer Client-Server-Verbindung sind für eine andere Client-Server-Verbindungen nicht sichtbar. D.h. ein Client kann die „non-persistent“ DataSets eines anderen Clients nicht sehen und auch nicht löschen. Ein „non-persistent“ DataSet kann nur von dem Client gelöscht werden, der ihn auch erstellt hat oder wird automatisch gelöscht, wenn die Verbindung dieses Clients unterbrochen wurde.

Die „persistent“ DataSets dagegen sind für alle Clients sichtbar und können auch zum späteren Zeitpunkt von einem anderen Client gelöscht werden. Allerdings ist davon abzuraten da der Ersteller-Client es vielleicht nicht mitbekommt und möglicherweise versucht, auf diesen DataSet weiterhin zuzugreifen. Die „persistent“ DataSets werden beim Verbindungsabbruch nicht automatisch gelöscht und verbleiben so im Datenmodell, bis sie explizit gelöscht werden.

Ein dynamisch erstelltes „persistent“ oder „non-persistent“ DataSet, dass z. B. von einem Report-Kontrollblock verwendet wird kann nicht gelöscht werden. D.h. solange der Kontrollblock Attribut: „DataSet“ diesen DataSet referenziert kann dieser DataSet nicht gelöscht werden. Wenn Sie einen solchen DataSet löschen wollen, dann muss das Attribut „DataSet“ einen anderen DataSet referenzieren oder die „DataSet“ Referenz muss gelöscht werden (ein Leerstring löscht z.B. die „DataSet“ Referenz).

Bei einem Client-Projekt und nach einem TwinCAT-Neustart oder einem Reset der TwinCAT-SPS werden alle „persistent“ und „non-persistent“ DataSets auf der Clientseite automatisch gelöscht. Auf der Serverseite verbleiben aber möglicherweise die „persistent“ DataSets erhalten da sie nicht explizit gelöscht werden konnten.

Die statischen DataSets sind im Grunde auch „persistent“ DataSets mit dem Unterschied zu den dynamisch erzeugten DataSets, dass sie nicht gelöscht werden können.

Für jeden DataSet, den Sie zum späteren Zeitpunkt (zur Laufzeit) dynamisch erstellen wollen, benötigen Sie eine Instanz des Funktionsbausteins: „FB_AcsiCommonDataSetClass“. Diese Instanzen werden während der SPS-Programmierung manuell dem SPS-Projekt bereits vorher hinzugefügt. Sie sind aber noch nicht mit dem Client-Datenmodell verknüpft. Die SPS-Applikation kann dann später zur Laufzeit diese Instanzen zur Erstellung der dynamischen DataSets auf der Serverseite verwenden. Erst dann werden diese Instanzen auch mit dem Datenmodell auf der Clientseite verknüpft. Die DataSet-Member können ebenfalls bereits im Voraus vorkonfiguriert werden. Die Methode „CreateDataSetReq“ benötigt als ersten Parameter: „ipDataSet“

einen Schnittstellenzeiger auf ein solches, vorkonfiguriertes DataSet Objekt. Der zweite Parameter: „ipLogicalNode“ bestimmt, ob ein „persistent“ oder „non-persistent“ DataSet erstellt werden soll. Nur die „persistent“ DataSets werden mit einem logischen Knoten verknüpft. In diesem Fall muss dieser Parameter gültig sein. Bei einem „non-persistent“ DataSet ist dieser Parameter Null. Beim Erfolg wird auf der Serverseite ein dynamisches DataSet erstellt und die vorkonfigurierte DataSet-Instanz mit dem Datenmodell auf der Clientseite verknüpft. Die Methode „DeleteDataSetReq“ benötigt als ersten Parameter: „ipDataSet“ einen Schnittstellenzeiger auf ein DataSet Objekt (welches bereits vorher dem Datenmodell hinzugefügt wurde). Beim Erfolg wird dann das DataSet auf der Serverseite aus dem Datenmodell gelöscht und auf der Clientseite die Verknüpfung des DataSets mit dem Datenmodell ebenfalls gelöscht.

Beispielprojekt

In dem zip-Archiv befindet sich eine ICD-Konfigurationsdatei. Diese Datei kann verwendet werden, um z.B. einen Server mit Hilfe einer Drittherstellersoftware zu simulieren oder mit TwinCAT Telecontrol Configurator ein neues oder modifiziertes TwinCAT Projekt zu erzeugen.

Im Funktionsbaustein FB_IEDClient und State 10 sind mehrere IF-Anweisungen aufgeführt, welche Befehle (Methodenaufrufe) des Client-Bausteins aktivieren. Jeder Befehl in der IF-Anweisung wird durch eine steigende Flanke an einer der booleschen Variablen aktiviert. Die booleschen Variablen sind im Deklarationsteil des Funktionsbausteins FB_IEDClient definiert. Zu Testzwecken kann der Benutzer in der Online-Ansicht die Werte der booleschen Variablen auf „TRUE“ setzen und die Befehlsabarbeitung auf diese Weise aktivieren. Danach wird zum State 11 gewechselt, wo in weiteren SPS-Zyklen die Methode „ipResult.Execute()“ so lange aufgerufen wird, bis die Befehlsabarbeitung abgeschlossen wurde. Dies ist dann der Fall, wenn „ipResult.IsBusy() FALSE zurück liefert.

Nach der erfolgreichen Abarbeitung eines Befehls wird die State-Machine in den State 0 versetzt. Falls mehrere der booleschen Variablen auf „TRUE“ gesetzt sind, dann wird die oberste IF-Anweisung mit dem jeweiligen Befehl zuerst abgearbeitet. In der folgenden Tabelle sind einige der booleschen Variablen aufgelistet. Zudem sind die Methodennamen und eine Beschreibung der Funktionalität enthalten.

Das Beispiel verwendet zwei vorkonfigurierte Instanzen des „FB_AcsiCommonDataSetClass“-Funktionsbausteins. Die Instanz: „fbNonPersistent“ um einen „non-persistent“ und die Instanz: „fbPersistent“ um einen „persistent“ DataSet zu erzeugen. Die Namen der DataSets wurden bereits im Deklarationsteil mit Hilfe der Eigenschaft „sObjectName“ auf „NonPersistentDataSet“ und „PersistentDataSet“ vorkonfiguriert.

```
...
fbNonPersistent: FB_AcsiCommonDataSetClass:=(sObjectName:='NonPersistentDataSet');
fbPersistent: FB_AcsiCommonDataSetClass:=(sObjectName:='PersistentDataSet');
...
```

In der IF-Anweisung, kurz vor der Aktivierung der „CreateDataSetReq“-Methode und wenn bereits noch nicht geschehen, werden die DataSet-Member den DataSets hinzugefügt. Dies kann aber auch in eine separate Routine ausgelagert werden die z.B. einmalig beim Programmstart aufgerufen wird.

```
...
ELSIF bCreateDataSet_NonPersistent THEN
  bCreateDataSet_NonPersistent:= FALSE;
  IF fbNonPersistent.nMembers = 0 THEN
    bSuccess:= fbNonPersistent.AddMember(ipData:=fbIED.IEDLD1.XCBR1.Pos.stVal, eFc:=E_AcsiFc.ST_);
  );
    bSuccess:= fbNonPersistent.AddMember(ipData:=fbIED.IEDLD1.LLN0.Mod_.stVal, eFc:=E_AcsiFc.ST_);
  );
  END_IF
  bSuccess:= fbConnection.CreateDataSetReq(ipDataSet:=fbNonPersistent, ipLogicalNode:=0, hUser:=0,
ipSink:=0, nInvokeID=>nInvokeID, ipResult=>ipResult);
  state:= SEL(bSuccess, 100, 11);
ELSIF bCreateDataSet_Persistent THEN
  bCreateDataSet_Persistent:= FALSE;
  IF fbPersistent.nMembers = 0 THEN
    bSuccess:= fbPersistent.AddMember(ipData:=fbIED.IEDLD1.MMXU1.TotW.mag.f, eFc:=E_AcsiFc.MX);
    bSuccess:= fbPersistent.AddMember(ipData:=fbIED.IEDLD1.LLN0.Beh.stVal, eFc:=E_AcsiFc.ST_);
  END_IF
  bSuccess:= fbConnection.CreateDataSetReq(ipDataSet:=fbPersistent, ipLogicalNode:=fbIED.IEDLD1.LLN0,
hUser:=0, ipSink:=0, nInvokeID=>nInvokeID, ipResult=>ipResult);
  state:= SEL(bSuccess, 100, 11);
...

```


| Variablenname | Methodenname | Beschreibung |
|------------------------------------|---------------------------|---|
| bCreateDataSet_NonPersistent | CreateDataSetReq | Eine steigende Flanke an dieser Variablen aktiviert den Befehl zur Erstellung eines dynamischen „non-persistent“ DataSets mit dem Namen: „NonPersistentDataSet“ (Objektreferenz: '@NonPersistent'). |
| bCreateDataSet_Persistent | CreateDataSetReq | Eine steigende Flanke an dieser Variablen aktiviert den Befehl zur Erstellung eines dynamischen „persistent“ DataSets mit dem Namen: „PersistentDataSet“ (Objektreferenz: 'IEDLD1/LLN0.PersistentDataSet'). Beim Erfolg wird der neue DataSet mit dem logischen Knoten „LLN0“ verknüpft. |
| bDeleteDataSet_NonPersistent | DeleteDataSetReq | Eine steigende Flanke an dieser Variablen aktiviert den Befehl zum Löschen eines dynamischen „non-persistent“ DataSets mit dem Namen: „NonPersistentDataSet“. |
| bDeleteDataSet_Persistent | DeleteDataSetReq | Eine steigende Flanke an dieser Variablen aktiviert den Befehl zum Löschen eines dynamischen „persistent“ DataSets mit dem Namen: „PersistentDataSet“. |
| bEnable_urcb101_NonPersistent | urcb101.Client.EnableReq | Eine steigende Flanke an dieser Variablen aktiviert die Berichtssteuerblockinstanz für ungepufferte Reports: „urcb101“. Die Reports sollen die DataSet-Member des zuvor dynamisch erstellten „non-persistent“ DataSets: „fbNonPersistent“ übertragen. Intern wird vorher noch das Attribut: „DataSet“ des Berichtssteuerblocks entsprechend gesetzt (der String referenziert den dynamisch erstellten DataSet). |
| bEnable_urcb101_Persistent | urcb101.Client.EnableReq | Eine steigende Flanke an dieser Variablen aktiviert die Berichtssteuerblockinstanz für ungepufferte Reports „urcb101“. Die Reports sollen die DataSet-Member des zuvor dynamisch erstellten „persistent“ DataSets: „fbPersistent“ übertragen. Intern wird vorher noch das Attribut: „DataSet“ des Berichtssteuerblocks entsprechend gesetzt (der String referenziert den dynamisch erstellten DataSet). |
| bEnable_urcb101_DS1 | urcb101.Client.EnableReq | Eine steigende Flanke an dieser Variablen aktiviert die Berichtssteuerblockinstanz für ungepufferte Reports „urcb101“. Die Reports sollen die DataSet-Member des statischen DataSets: „DS1“ übertragen. Intern wird vorher noch das Attribut: „DataSet“ des Berichtssteuerblocks entsprechend gesetzt (der String referenziert den statischen DataSet). |
| bDisable_urcb101 | urcb101.Client.DisableReq | Eine steigende Flanke an dieser Variablen deaktiviert die Berichtssteuerblockinstanz für ungepufferte Reports „urcb101“. Die Übertragung der Reports wird daraufhin gestoppt. |
| bSetDataValues_urcb101_DatSet | SetDataValuesReq | Solange „DataSet“ einen dynamisch erstellten DataSet referenziert, kann dieser DataSet nicht wieder gelöscht werden. Um einen dynamisch erstellten DataSet löschen zu können darf er von keinem Kontrollblock verwendet werden und auch von dem „DataSet“ nicht referenziert werden. Eine steigende Flanke an dieser Variablen beschreibt den Attributwert: „IEDLD1/LLN0.RP.urcb101.DatSet“ mit einem Leerstring. |
| bGetDataSetDirectory_NonPersistent | GetDataSetDirectoryReq | Eine steigende Flanke an dieser Variablen aktiviert den Befehl zum Lesen der DataSet-Member Konfiguration des zuvor dynamisch erstellten „non-persistent“ DataSets: „fbNonPersistent“. Die DataSet-Member werden dabei enumeriert. |
| bGetDataSetDirectory_Persistent | GetDataSetDirectoryReq | Eine steigende Flanke an dieser Variablen aktiviert den Befehl zum Lesen der DataSet-Member Konfiguration des zuvor dynamisch erstellten „persistent“ DataSets: „fbPersistent“. Die DataSet-Member werden dabei enumeriert. |
| bGetDataSetDirectory_DS1 | GetDataSetDirectoryReq | Eine steigende Flanke an dieser Variablen aktiviert den Befehl zum Lesen der DataSet-Member Konfiguration des statischen DataSets: „DS1“. Die DataSet-Member werden dabei enumeriert. |

Kleiner Test

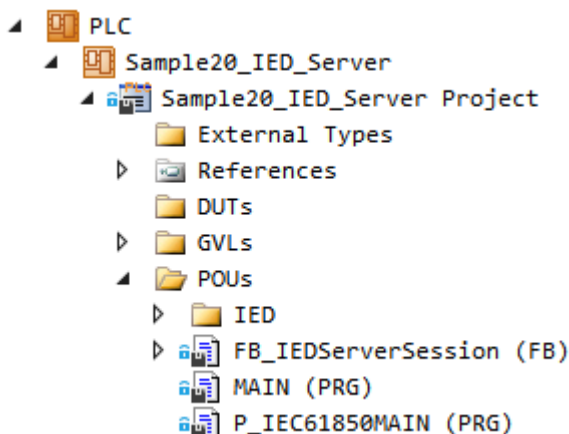
Nach dem Programmstart, wenn die Verbindung zum Server steht, schreiben sie zuerst den Wert TRUE in die boolesche Variable: „bCreateDataSet_NonPersistent“. Beim Erfolg wird ein neues dynamisches „non-persistent“ DataSet erstellt. Dieser DataSet kann dann von der Berichtssteuerblockinstanz für ungepufferte Reports verwendet werden. Wenn sie den Wert TRUE in die Variable „bEnable_urcb101_NonPersistent“ schreiben dann aktivieren Sie das Versenden der Reports mit den „non-persistent“ DataSet-Member Daten. Stoppen sie das Versenden der Reports, indem Sie den Wert TRUE in die Variable: „bDisable_urcb101“ schreiben. Der dynamische „non-persistent“ DataSet wird aber immer noch von dem Attribut: „DataSet“ der Berichtssteuerblockinstanz „urcb101“ verwendet und lässt sich noch nicht löschen. Schreiben Sie den Wert TRUE in die boolesche Variable „bSetDataValues_urcb101_DatSet“. Jetzt kann der dynamische „non-persistent“ DataSet gelöscht werden. Setzen sie dafür den Wert TRUE in die Variable: „bDeleteDataSet_NonPersistent“.

7.15 Allgemeine Server - Projektstruktur

Alle Server-Beispielprojekte folgen, ähnlich wie die Client-Beispielprojekte, ebenfalls einer allgemeinen Struktur. Im Gegensatz zum Client wird beim Server die Verbindung nicht aktiv Serverseits aufgebaut. Nach dem Programmstart wartet der Server auf einen Verbindungswunsch eines Clients und akzeptiert oder lehnt diesen ab. Außerdem muss es für den Server möglich sein, Verbindungen zu mehreren Clients aufzubauen.

Daraus ergibt sich eine andere Server-Projektstruktur als beim Client. Die Grundstruktur der TwinCAT IEC 61850 Server-Beispielprojekte basiert auf der automatisch vom TwinCAT Telecontrol Configurator generierten SPS-Projektstruktur. Der TwinCAT Telecontrol Configurator verwendet bei der Generierung der Solution wiederum das „TwinCAT XAE Project (XML format)“ als Vorlage. Die in den Beispielen kommunizierten Datenmodelle spiegeln unterschiedliche IEC 61850 Server wider. Sie unterscheiden sich von Beispiel zu Beispiel und liegen dem Quellcode als ICD-Dateien bei. Die ICD-Dateien können auch von einer Third-Party-Software zur Simulation verwendet werden.

Aufbau eines TwinCAT IEC 61850-SPS-Server-Projekts:



Der generierte Solution-Name (wenn nicht anders angegeben) entspricht dem TwinCAT Telecontrol Configurator Projektnamen. Der automatisch generierte TwinCAT SPS-Projektname (wenn nicht anders angegeben) hat wiederum folgenden Aufbau: „[Projektname]_[IEDName]_Server“.

Standardmäßig besitzt jedes Beispielprojekt einen „DUTs“, „GVLs“ und „POUs“ Ordner. In dem „GVLs“-Ordner ist eine globale Variablenliste (GVL) mit dem Namen: „TcTelecontrol“ hinterlegt. In dieser globalen Variablenliste (siehe Quellcode unten) werden folgende Funktionsbausteine instanziiert und initialisiert:

- Eine Server-Bausteininstanz des Typs: „FB_iec61850ServerClass“ (Server Konfigurationseinstellungen und Verbindungsmanagement).
- Mindestens eine Instanz des Server-Session-Bausteins des Typs: „FB_[IEDName]ServerSession[1..n]“ (Verbindungsmanagement einer einzelnen Verbindung und Datenaustausch mit einem Client). Fügen Sie an dieser Stelle manuell weitere Instanzen des Server-Session-Bausteins hinzu, falls der Server gleichzeitig mit mehr als nur einem Client kommunizieren soll.
- Eine IED-Datenmodell-Bausteininstanz des Typs: „FB_IED_[IEDName]“.
- Optional (abhängig von der GOOSE-Publisher-Konfiguration): Eine oder mehrere Bausteininstanzen des Typs: „FB_[IEDName]Gse“ für GOOSE-Kommunikation und GSE-Management.

Außerdem wird dort auch die, während der Codegenerierung verwendete Codeversion und die verwendete Version des TwinCAT Telecontrol Configurator abgelegt.

Namensraum: TcTelecontrol

Typ: Globale Variablenliste (GVL)

```
VAR_GLOBAL
    ipCreator          : I_AcsiCodeCreatorClass := GVL_AcsiVars.Creator.SetCodeRev(codeRev:=2
).SetGuiVer(major:=1, minor:=1, build:=94, revision:=1);
    fb[IEDName]       : FB_IED_[IEDName];
    fb[IEDName]Server : FB_iec61850ServerClass := (ipIED:=fb[IEDName], settings:=(bEnable:=T
RUE, sLocalHost:='127.0.0.1'));
    fb[IEDName]ServerSession1 : FB_[IEDName]ServerSession := (fbConnection:=(ipServer:=fb[IEDName]Se
rver, settings:=(bEnable:=TRUE)));
    fb[IEDName]Gse     : FB_[IEDName]Gse := (fbAdapter:=(ipIED:=fb[IEDName], settings:=(sMult
icastAddr:='01-0C-CD-01-00-00', eDispatchMode:=E_GseDispatchMode.NonPromiscuous)));
END_VAR
```

Im „POUs“-Ordner befindet sich ein weiterer Ordner: „[IEDName]“, der die gesamte hierarchische Struktur des IED-Datenmodells als Funktionsbausteine beinhaltet. In diesem Ordner befindet sich auch der IED-Datenmodell-Baustein, der in der globalen Variablenliste instanziiert wird und bereits erwähnt wurde. Auf der gleichen Ebene liegt der Server-Session-Baustein: „FB_[IEDName]ServerSession“, welcher den Verbindungsaufbau und den Datenaustausch mit einem IEC 61850 Client realisiert.

Das TwinCAT SPS-Projekt enthält standardmäßig ein „MAIN“-Programm. Dieses wird zyklisch von einer TwinCAT Task aufgerufen und ruft wiederum das Programm „P_IEC61850MAIN“ auf. Das Programm „P_IEC61850MAIN“ kapselt den Aufruf des Server-, Server-Session- und des optionalen Gse-Bausteins, trennt die IEC 61850-Kommunikation von dem restlichen SPS-Maschinenprogramm.

```
PROGRAM MAIN
VAR
END_VAR

P_IEC61850MAIN();

PROGRAM P_IEC61850MAIN
VAR
END_VAR

fb[IEDName]Server.Execute();
fb[IEDName]ServerSession1();
fb[IEDName]Gse();
```

In dem FB_[IEDName]ServerSession-Baustein befindet sich eine State-Machine, von der die grundlegenden Zustände in jedem Server-Beispiel genutzt werden (siehe Quellcode unten). Der Server-Session-Baustein ist verantwortlich für den Verbindungsaufbau und Datenaustausch zu einem einzelnen Client. Bei mehreren gleichzeitigen Clientverbindungen werden mehrere Instanzen von diesem Baustein benötigt und instanziiert.

State 0 (Init state): Die State-Machine befindet sich in diesem Zustand, sobald das SPS-Programm gestartet wurde. Hier (und im State 1) werden Befehle zur Verwaltung einer -Server—Client-Verbindung behandelt. Im Regelfall verbleibt der Server-Session-Baustein in diesem Zustand und wartet auf einen Verbindungswunsch eines Clients. Der Server baut dann eine Verbindung auf, sobald der Verbindungswunsch erkannt wurde. Dies geschieht automatisch und benötigt keine weiteren Befehle seitens der SPS-Applikation. Falls der Server-Verbindung zum Client hergestellt hat, wird die State-Machine in den Zustand der Datenübertragung wechseln (State 10).

Es ist außerdem möglich, eine bereits hergestellte Verbindung zu schließen. Gesteuert wird dies über zwei boolesche Variablen. Diese Variablen, wenn sie gesetzt sind, aktivieren die entsprechenden Befehle (in diesem Fall sind es einmalige Methodenaufrufe am Server-Session-Baustein).

- **_bAbort:** Ruft die Methode „AbortReq“ auf, welche den Befehl zum Abbruch der Server-Verbindung zum Client aktiviert.
- **_bDisconnect:** Ruft die Methode „ReleaseReq“ auf, welche den Befehl zum geregelten Abbau einer bestehenden Server-Verbindung zum Client aktiviert. Im Regelfall ist es aber der Client, der seine aktive Verbindung zum Server beenden sollte.

Die zuvor aufgezählten Methoden benötigen für die Abarbeitung länger als einen SPS-Zyklus. Aus diesem Grund wechselt die State-Machine in einen Warte-Zustand (State 1), in dem auf die Beendigung des aktivierten Befehls gewartet wird.

State 1 (Wait State): In diesem Zustand verweilt der Server-Session-Baustein, bis die Befehlsabarbeitung zur Verwaltung der Server-Client-Verbindung nicht mehr beschäftigt (busy) ist. Solange die Verbindung abgebaut oder abgebrochen wird, befindet sich die Statemachine im State 1. Sobald der Befehl erfolgreich abgearbeitet wurde, wird die Statemachine zurück in den State 0 (Init State) versetzt.

State 10 (Data exchange): Wenn die Statemachine in diesem Zustand ist, dann ist die Server-Verbindung zum Client bereits erfolgreich hergestellt worden. Der Server-Session-Baustein ist bereit für die Datenübertragung zum Server. Die Server-Datenübertragung zum Client geschieht automatisch im Hintergrund und muss nicht seitens der SPS-Applikation initiiert werden. Es ist der Client, der durch einen Request eine Datenübertragung im Server auslöst.

Bei einer aktiven Verbindung wechselt die Statemachine zwischen dem State 0 und 10. Die Statemachine wird wieder in den State 0 versetzt, um auf Änderungen im Client-Server-Verbindungsstatus zu reagieren und sie im State 0 zu behandeln.

State 100 (Error state): Sobald ein Fehler bei der Aktivierung oder Abarbeitung eines Befehls auftritt, wird die Statemachine in den State 100 versetzt. Hier wird der Fehler geloggt und die Statemachine zurück in den State 0 versetzt.

```

FUNCTION_BLOCK FB_[IEDName]ServerSession
VAR_INPUT
    fbConnection      : FB_iec61850ConnectionClass := (ipAbortInd:=THIS^, ipAssociateInd:=THIS^, ipRe
leaseInd:=THIS^);
END_VAR
VAR
    _bAbort           : BOOL;
    _bDisconnect      : BOOL;
    state             : BYTE;
    eState             : E_AsyncEnvironmentState;
    bBusy              : BOOL;
    bSuccess           : BOOL;
    ipResult           : I_AsyncServiceResultClass;
    sLastErrorResult  : T_MaxString;
    fbAbortReason     : FB_ServiceErrorClass := (stError:=SUCCESS_EVENT);
    sLastAbortReason  : T_MaxString;
END_VAR

fbConnection.Execute();
eState:= fbConnection.eState;

CASE state OF
    0:
        IF _bAbort THEN
            _bAbort:= FALSE;
            bSuccess:= fbConnection.AbortReq(ipReason:=fbAbortReason, ipSink:=0, ipResult=>ipResult)
;
            state:= SEL(bSuccess, 100, 1);
        ELSIF eState = E_AsyncEnvironmentState.Established AND _bDisconnect THEN
            _bDisconnect:= FALSE;
            bSuccess:= fbConnection.ReleaseReq(ipSink:=0, ipResult=>ipResult);
            state:= SEL(bSuccess, 100, 1);
        ELSIF eState = E_AsyncEnvironmentState.Established THEN
            state:= 10;
        END_IF
        _bDisconnect:= FALSE;
    1:
        IF ipResult <> 0 THEN
            ipResult.Execute();
            IF NOT (bBusy:=ipResult.IsBusy()) THEN
                state:= SEL(ipResult.IsCompleted(), 100, 0);
            END_IF
        END_IF
    10:
        state:= 0;
    100:
        state:= 0;
        IF ipResult <> 0 THEN
            sLastErrorResult:= ipResult.Dump();
        END_IF
END_CASE
END_FUNCTION_BLOCK

```

GOOSE-Publisher (optional)

TwinCAT Telecontrol Configurator kann in einem Server-Projekt, während der SPS-Code Generierung, den SPS-Code für einen GOOSE-Publisher ebenfalls generieren (siehe Codebeispiel unten). Dies ist aber nur dann möglich, wenn der Anwender vorher die GOOSE-Komponenten wie z. B. GoCBs (goose control

blocks) im TwinCAT Telecontrol Configurator erstellt oder aus einer SCL-Datei (z. B. ICD-Datei) importiert hat. Standardmäßig wird bei der Codegenerierung ein Baustein mit dem Namen: „FB_[IEDName]Gse“ instanziiert und der globalen Variablenliste „TcTelecontrol“ hinzugefügt. Dieser Baustein stellt die Verbindung zwischen einem Netzwerkadapter des TwinCAT Steuerungsrechners, dem IED-Datenmodell und der GOOSE-Konfiguration in den GoCBs her. Die GoCBs werden in dem IED-Datenmodell (normalerweise im LLN0) instanziiert. Jeder GoCB besitzt ein Baustein-Unterelement mit dem Namen: „Publisher“. Über die „Publisher“-Methodenaufrufe kann das Publishing aus dem SPS-Code gestartet („Start“) oder gestoppt werden („Stop“). Defaultmäßig wird das Publishing beim SPS-Programmstart für alle GoCBs gestartet. Dies wird durch die mit „TRUE“ initialisierte „bStart“-Variable gesteuert. Über eine steigende Flanke an der „bStop“-Variablen kann das Publishing für alle GoCBs gestoppt werden. Die über diese Methoden abgesetzten Publisher-Befehle werden sofort ausgeführt, ohne Wartezyklen oder weitere States die nötig sind, um die Befehlsabarbeitung abzuschließen. Der hier beschriebene „Publisher“ liest die Konfiguration und aktualisiert zwar den Status des GoCBs (Attribute „GoEna“ wird z. B. auf „TRUE“ oder „FALSE“ gesetzt), er nutzt aber nicht die Client-Server-Dienste wie „SetGoCBValues“ oder „GetGoCBValues“ um den Publisher zu starten oder zu stoppen. D.h. der generierte Code implementiert bereits einen Publisher der z. B. im ersten SPS-Zyklus oder aus dem SPS-Code zum beliebigen Zeitpunkt gestartet bzw. gestoppt werden kann. Die benötigten GoCB-Konfigurationseinstellungen (GoCB-Attributwerte) können über Initialisierungswerte vorgenommen werden. Die Konfiguration der GoCBs kann aber bereits in dem TwinCAT Telecontrol Configurator geschehen. Die Initialisierungswerte werden dann während der Codegenerierung automatisch generiert und zugewiesen. Wenn der Publisher gestartet wurde und die Konfiguration des GoCBs und des Netzwerkadapters eine Übereinstimmung ergibt, dann beginnt der Publisher sofort mit dem Versenden der ersten GOOSE-Frames (inklusive der Framewiederholungen). Die „Update“-Methode hat eine besondere Bedeutung. Jedes Mal wenn sie aufgerufen wird, versendet der „Publisher“ sofort ein neues GOOSE-Frame (und automatisch neue Framewiederholungen). D.h. die Applikation kann alle relevanten GOOSE-Dataset-Daten im IED-Datenmodell zuerst setzen und dann das Versenden der geänderten Dataset-Daten mit dem „Update“-Methodenaufwurf veranlassen. Die „Execute“-Methode muss in der übrigen Zeit zyklisch aufgerufen werden. Sie ist für das Versenden der Framewiederholungen und für die Aktualisierung der Statusinformation im GoCB verantwortlich.

```

FUNCTION_BLOCK FB_[IEDName]Gse IMPLEMENTS I_GseLinkStatusEventSink
VAR_INPUT
    fbAdapter      : FB_GseAdapterClass := (ipLinkStatus:=THIS^);
END_VAR
VAR
    eLinkStatus   : E_GseLinkStatus;
    bSuccess      : BOOL;
    ipError       : I_ServiceErrorClass;
    bStart        : BOOL := TRUE;
    bStop         : BOOL;
    bUpdate       : BOOL;
END_VAR

bSuccess:= fbAdapter.Execute(ipError=>ipError);

IF bStart THEN
    bStart:= FALSE;
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb01.Publisher.Start(ipAdapter:=fbAdapter, ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb02.Publisher.Start(ipAdapter:=fbAdapter, ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb03.Publisher.Start(ipAdapter:=fbAdapter, ipError=>ipError);
ELSIF bStop THEN
    bStop:= FALSE;
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb01.Publisher.Stop(ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb02.Publisher.Stop(ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb03.Publisher.Stop(ipError=>ipError);
ELSIF bUpdate THEN
    bUpdate:= FALSE;
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb01.Publisher.Update(ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb02.Publisher.Update(ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb03.Publisher.Update(ipError=>ipError);
ELSE
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb01.Publisher.Execute(ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb02.Publisher.Execute(ipError=>ipError);
    bSuccess:= fb[IEDName].IEDLD1.LLN0.gocb03.Publisher.Execute(ipError=>ipError);
END_IF

```

Der GSE-Baustein implementiert die „I_GseLinkStatusEventSink“-Schnittstelle. Die Methode: „OnLinkStatusChange“ gehört zu dieser Schnittstellenimplementierung und wird immer dann aufgerufen, wenn sich der Status der Netzwerkverbindung (am Netzwerkadapter) ändert. Die SPS-Applikation kann z. B. den Netzwerkverbindungsstatus via „eLinkStatus“-Variable abfragen oder überprüfen.

```

METHOD OnLinkStatusChange
VAR_INPUT
    ipAdapter : I_GseAdapterClass;
    eStatus   : E_GseLinkStatus;
END_VAR
VAR
END_VAR
eLinkStatus:= eStatus;

```

Im Projektbaum unter dem Zweig I/O-Device finden Sie eine Netzwerkadapterinstanz mit dem Namen „GSE (RT-Ethernet Adapter)“. Diese Adapterinstanz muss entsprechend konfiguriert werden, d.h. die I/O-Konfiguration muss an die vorhandene Hardware und auf die Zielplattform, auf der das Projekt laufen soll, angepasst werden.

Eine erneute I/O-Konfiguration ist auch dann nötig, wenn Sie die Zielplattform wechseln. Diese Konfiguration muss manuell in TwinCAT XAE vorgenommen werden. Zusätzlich zu der I/O-Konfiguration des Netzwerkadapters muss ein Link zwischen dem Netzwerkadapter und den SPS-Bausteinen für die Goose-Kommunikation hergestellt werden. Mit dem Link können die vom Netzwerkadapter empfangenen Daten an die Instanz des Funktionsbausteins „FB_[IEDName]Gse“ weitergeleitet werden. In umgekehrter Richtung kann die Instanz des Funktionsbausteins „FB_[IEDName]Gse“ die zu sendenden Daten an den Netzwerkadapter weiterleiten.

Hier finden Sie weitere Informationen: [RT – Ethernet Adapter Konfiguration](#) [► 37].

7.16 Server - Basis Beispielprojekt

Dieses Beispiel zeigt die grundlegende TwinCAT Implementierung der Client-Server-Kommunikation eines TwinCAT IEC 61850 Servers. Die GOOSE-Komponenten und Reporting sind der Einfachheit halber nicht in diesem Beispiel enthalten. Diese und andere Funktionalitäten werden in separaten Beispielen behandelt. Der hier implementierte Server hat eine Default-Host-Adresse: „127.0.0.1“ und Default-Portnummer: 102. Diese Werte können im SPS-Beispielcode angepasst werden. Die SPS-Entwicklungsumgebung wird beim Übersetzen eine Warnung mit einem Hinweis dazu ausgeben. Ein Doppelklick auf diese Meldung leitet Sie zu der entsprechenden Stelle im SPS-Code.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1031/TF6510_TC3_IEC61850/Resources/10698184715.zip

Das hier beschriebene Beispiel nutzt die Statemachine, welche in dem Kapitel „[Allgemeine Server - Projektstruktur](#) [► 628]“ beschrieben ist. Die States: 0,1,10 und 100 sind identisch zu der dort beschriebenen Statemachine. Andere States wurden für das Beispiel modifiziert oder auch neue States hinzugefügt.

7.17 Server – GOOSE Publisher (GetGoCBDataValues, SetGoCBDataValues)

In diesem Beispiel kann der GOOSE-Steuerblock seitens Clients mit Hilfe der Dienste GetGoCBDataValues und SetGoCBDataValues kontrolliert werden.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1031/TF6510_TC3_IEC61850/Resources/10698177035.zip

Das hier beschriebene Beispiel nutzt die Statemachine, welche in dem Kapitel „[Allgemeine Server - Projektstruktur](#) [► 628]“ beschrieben ist. Die States: 0,1,10 und 100 sind identisch zu der dort beschriebenen Statemachine. Andere States wurden für das Beispiel modifiziert oder auch neue States hinzugefügt.

Im Projektbaum unter dem Zweig I/O-Device finden Sie eine Netzwerkadapterinstanz mit dem Namen „GSE (RT-Ethernet Adapter)“. Diese Adapterinstanz muss entsprechend konfiguriert werden, d.h. die I/O-Konfiguration muss an die vorhandene Hardware und auf die Zielplattform, auf der das Projekt laufen soll, angepasst werden.

Eine erneute I/O-Konfiguration ist auch dann nötig, wenn Sie die Zielplattform wechseln. Diese Konfiguration muss manuell in TwinCAT XAE vorgenommen werden. Zusätzlich zu der I/O-Konfiguration des Netzwerkadapters muss ein Link zwischen dem Netzwerkadapter und den SPS-Bausteinen für die Goose-Kommunikation hergestellt werden. Mit dem Link können die vom Netzwerkadapter empfangenen Daten an

die Instanz des Funktionsbausteins: „FB_[IEDName]Gse“ weitergeleitet werden. In umgekehrter Richtung kann die Instanz des Funktionsbausteins „FB_[IEDName]Gse“ die zu sendenden Daten an den Netzwerkadapter weiterleiten.

Hier finden Sie weitere Informationen: [RT – Ethernet Adapter Konfiguration \[► 37\]](#).

7.18 Server - GOOSE Publisher (ohne Client-Server-Kommunikation)

In einem Server-Projekt wird von dem TwinCAT Telecontrol Configurator standardmäßig auch ein Server-Baustein und einer oder mehrere Instanzen des ServerSession-Bausteins generiert (siehe: „[Allgemeine Server-Projektstruktur](#)“ [► 628]). Der Server und ServerSession-Baustein kann z. B. dafür sorgen, dass der Publisher vom Client aus via Client-Server-Dienste wie „GetGoCBDataValues“ oder „SetGoCBDataValues“ aktiviert/deaktiviert oder die GoCBs konfiguriert werden können. In manchen Fällen soll aber auf eine Client-Server-Kommunikation mit dem Gerät verzichtet werden und nur ein „reiner“ GOOSE Publisher implementiert werden. Ein solcher Publisher kann nach dem SPS-Programmstart automatisch mit dem Versenden der GOOSE-Nachrichten beginnen. Dieses Beispiel zeigt die Implementierung eines Publishers, aber ohne den Client-Server-Kommunikationsteil. Die nicht benötigten Teile des Server-Projekts wurden in diesem Beispiel bewusst entfernt.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1031/TF6510_TC3_IEC61850/Resources/7280992907.zip

Namensraum: TcTelecontrol

Typ: Globale Variablenliste (GVL)

In der globalen Variablenliste wird eine IED-Datenmodell-Bausteininstanz („fbIED“) und eine oder mehrere GSE-Bausteininstanzen für GOOSE-Kommunikation und GSE-Management („fbIEDGse“) benötigt. Dieses Beispiel verwendet nur einen Netzwerkadapter für die GOOSE-Kommunikation und drei GoCBs im IED-Datenmodell.

```
VAR_GLOBAL
    ipCreator : I_AcsiCodeCreatorClass := GVL_AcsiVars.Creator.SetCodeRev(codeRev:=2).SetGuiVer(major:=1, minor:=0, build:=93, revision:=10);
    fbIED      : FB_IED_IED;
    fbIEDGse  : FB_IEDGse := (fbAdapter:=(ipIED:=fbIED, settings:=(sMulticastAddr:='01-0C-CD-01-00-01', eDispatchMode:=E_GseDispatchMode.NonPromiscuous)));
END_VAR
```

Das „MAIN“-Programm wird zyklisch von einer TwinCAT Task aufgerufen und muss nur das Programm „P_IEC61850MAIN“ aufrufen. „P_IEC61850MAIN“ ruft wiederum den GSE-Baustein auf, der für das Versenden der GOOSE-Nachrichten und das Mapping der empfangenen Daten vom IED-Datenmodell in die GOOSE-Nachrichten zuständig ist.

```
PROGRAM MAIN
VAR
END_VAR

P_IEC61850MAIN();

PROGRAM P_IEC61850MAIN
VAR
END_VAR

fbIEDGse();
```

In der Beispielimplementierung des GSE-Bausteins wird der Publishing-Prozess bei allen drei GoCBs nach dem SPS-Start automatisch gestartet. Eine steigende Flanke an der „bStop“-Variablen stoppt den Publishing-Prozess der GoCB's. Die Wertänderungen einiger GOOSE-Daten werden im Beispiel simuliert. Dabei werden alle 5s einige Werte der GoCB-DataSet-Member modifiziert und an der Variablen „bUpdate“ eine steigende Flanke generiert. Über diese steigende Flanke werden die Publisher veranlasst eine neue GOOSE-Nachricht zu generieren und zu versenden. Die SPS-Applikation muss sich um die GOOSE-Nachrichtwiederholungen nicht kümmern. Dies erfolgt automatisch.

```
FUNCTION_BLOCK FB_IEDGse IMPLEMENTS I_GseSystemClockEventSink, I_GseLinkStatusEventSink
VAR_INPUT
    fbAdapter      : FB_GseAdapterClass := (ipSystemClock:=THIS^, ipLinkStatus:=THIS^);
END_VAR
VAR
    eLinkStatus    : E_GseLinkStatus;
```

```

bSuccess      : BOOL;
ipError       : I_ServiceErrorClass;
bStart        : BOOL := TRUE;
bStop         : BOOL;
bUpdate       : BOOL;

bSimulation   : BOOL := TRUE;
tSimulation   : TIME := T#5S;
fbUpdateTimer : TON;

bSync         : BOOL := TRUE;
tSync         : T_UtcTime := String_TO_UtcTime(in:='UT#2019-07-12-12:00:00.000000000|000|3');
fbClock       : FB_GseSystemClock;
END_VAR

fbUpdateTimer(IN:=bSimulation, PT:=tSimulation);
IF fbUpdateTimer.Q THEN
  fbUpdateTimer(IN:=FALSE);
  fbUpdateTimer(IN:=bSimulation);

  fbIED.IEDLD1.LEDGGIO1.SPCS01.stVal.bValue:= NOT fbIED.IEDLD1.LEDGGIO1.SPCS01.stVal.bValue;
  fbIED.IEDLD1.LEDGGIO2.SPCS01.stVal.bValue:= NOT fbIED.IEDLD1.LEDGGIO2.SPCS01.stVal.bValue;
  fbIED.IEDLD1.LEDGGIO3.SPCS01.stVal.bValue:= NOT fbIED.IEDLD1.LEDGGIO3.SPCS01.stVal.bValue;
  fbIED.IEDLD1.LEDGGIO4.SPCS01.stVal.bValue:= NOT fbIED.IEDLD1.LEDGGIO4.SPCS01.stVal.bValue;
  fbIED.IEDLD1.LEDGGIO5.SPCS01.stVal.bValue:= NOT fbIED.IEDLD1.LEDGGIO5.SPCS01.stVal.bValue;
  fbIED.IEDLD1.LEDGGIO6.SPCS01.stVal.bValue:= NOT fbIED.IEDLD1.LEDGGIO6.SPCS01.stVal.bValue;
  fbIED.IEDLD1.LEDGGIO7.SPCS01.stVal.bValue:= NOT fbIED.IEDLD1.LEDGGIO7.SPCS01.stVal.bValue;
  fbIED.IEDLD1.LEDGGIO8.SPCS01.stVal.bValue:= NOT fbIED.IEDLD1.LEDGGIO8.SPCS01.stVal.bValue;

  fbIED.IEDLD1.MMXU1.TotW.mag.f.fValue:= fbIED.IEDLD1.MMXU1.TotW.mag.f.fValue + 0.1;

  IF fbIED.IEDLD1.XCBR1.Pos.stVal.eValue = E_AcsiDbpos.On THEN
    fbIED.IEDLD1.XCBR1.Pos.stVal.eValue:= E_AcsiDbpos.Off;
  ELSE
    fbIED.IEDLD1.XCBR1.Pos.stVal.eValue:= E_AcsiDbpos.On;
  END_IF
  fbIED.IEDLD1.XCBR1.Pos.q.OldData:= NOT fbIED.IEDLD1.XCBR1.Pos.q.OldData;
  fbIED.IEDLD1.XCBR1.Pos.t.SecondSinceEpoch:= fbIED.IEDLD1.XCBR1.Pos.t.SecondSinceEpoch + TIME#1S;

  bUpdate:= TRUE;
END_IF

IF bSync THEN
  bSync:= FALSE;
  bSuccess:= fbClock.SetToUtcTime(in:=tSync);
ELSE
  fbClock.Execute();
END_IF

bSuccess:= fbAdapter.Execute(ipError=>ipError);

IF bStart THEN
  bStart:= FALSE;
  bSuccess:= fbIED.IEDLD1.LLN0.gocb01.Publisher.Start(ipAdapter:=fbAdapter, ipError=>ipError);
  bSuccess:= fbIED.IEDLD1.LLN0.gocb02.Publisher.Start(ipAdapter:=fbAdapter, ipError=>ipError);
  bSuccess:= fbIED.IEDLD1.LLN0.gocb03.Publisher.Start(ipAdapter:=fbAdapter, ipError=>ipError);
ELSIF bStop THEN
  bStop:= FALSE;
  bSuccess:= fbIED.IEDLD1.LLN0.gocb01.Publisher.Stop(ipError=>ipError);
  bSuccess:= fbIED.IEDLD1.LLN0.gocb02.Publisher.Stop(ipError=>ipError);
  bSuccess:= fbIED.IEDLD1.LLN0.gocb03.Publisher.Stop(ipError=>ipError);
ELSIF bUpdate THEN
  bUpdate:= FALSE;
  bSuccess:= fbIED.IEDLD1.LLN0.gocb01.Publisher.Update(ipError=>ipError);
  bSuccess:= fbIED.IEDLD1.LLN0.gocb02.Publisher.Update(ipError=>ipError);
  bSuccess:= fbIED.IEDLD1.LLN0.gocb03.Publisher.Update(ipError=>ipError);
ELSE
  bSuccess:= fbIED.IEDLD1.LLN0.gocb01.Publisher.Execute(ipError=>ipError);
  bSuccess:= fbIED.IEDLD1.LLN0.gocb02.Publisher.Execute(ipError=>ipError);
  bSuccess:= fbIED.IEDLD1.LLN0.gocb03.Publisher.Execute(ipError=>ipError);
END_IF

```

Der GSE-Baustein implementiert die „I_GseSystemClockEventSink“-Schnittstelle. Die Methode: „OnGetSystemTime“ gehört zu dieser Schnittstellenimplementierung und wird immer dann aufgerufen, wenn ein Publisher für eine GOOSE-Nachricht einen neuen Zeitstempel benötigt. Die SPS-Applikation könnte z. B. einen eigenen Zeitstempel, der vielleicht von einer GPS-Uhr stammt, der GOOSE-Nachricht übergeben. Im Beispiel wird eine einfache Software-Demo-Uhr für die Zeitstempelung verwendet.

```

METHOD OnGetSystemTime : BOOL
VAR_INPUT
  ipAdapter : I_GseAdapterClass;

```



```

END_VAR
VAR_OUTPUT
    tT      : T_UtcTime;
END_VAR
VAR
END_VAR
END_VAR

OnGetSystemTime:= fbClock.OnGetSystemTime(ipAdapter:=ipAdapter, tT=>tT);

```

Der GSE-Baustein implementiert die „I_GseLinkStatusEventSink“-Schnittstelle. Die Methode: „OnLinkStatusChange“ gehört zu dieser Schnittstellenimplementierung und wird immer dann aufgerufen, wenn sich der Status der Netzwerkverbindung (am Netzwerkadapter) ändert. Die SPS-Applikation kann z. B. den Netzwerkverbindungsstatus via „eLinkStatus“-Variable abfragen oder überprüfen.

```

METHOD OnLinkStatusChange
VAR_INPUT
    ipAdapter : I_GseAdapterClass;
    eStatus   : E_GseLinkStatus;
END_VAR
VAR
END_VAR
END_VAR

eLinkStatus:= eStatus;

```

Im Projektbaum unter dem Zweig I/O-Device finden Sie eine Netzwerkadapterinstanz mit dem Namen „GSE (RT-Ethernet Adapter)“. Diese Adapterinstanz muss entsprechend konfiguriert werden, d.h. die I/O-Konfiguration muss an die vorhandene Hardware und auf die Zielplattform, auf der das Projekt laufen soll, angepasst werden.

Eine erneute I/O-Konfiguration ist auch dann nötig, wenn Sie die Zielplattform wechseln. Diese Konfiguration muss manuell in TwinCAT XAE vorgenommen werden. Zusätzlich zu der I/O-Konfiguration des Netzwerkadapters muss ein Link zwischen dem Netzwerkadapter und den SPS-Bausteinen für die Goose-Kommunikation hergestellt werden. Mit dem Link können die vom Netzwerkadapter empfangenen Daten an die Instanz des Funktionsbausteins: „FB_[IEDName]Gse“ weitergeleitet werden. In umgekehrter Richtung kann die Instanz des Funktionsbausteins „FB_[IEDName]Gse“ die zu sendenden Daten an den Netzwerkadapter weiterleiten.

Hier finden Sie weitere Informationen: [RT – Ethernet Adapter Konfiguration \[► 37\]](#).

7.19 Server - Direct Control with normal security

Dieses Beispiel zeigt die Implementierung von `ctlModel := 1` (Direct control with normal security, direct-operate) in einem Server für folgende CDCs:

- SPC (controllable single point)
- DPC (controllable double point)
- APC (controllable analogue process value)
- BAC (binary controlled analogue process value)
- BSC (binary controlled step position information)
- ENC (controllable enumerated status)
- INC (controllable integer status)
- ISC (integer controlled step position information)

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1031/TF6510_TC3_IEC61850/Resources/10698251403.zip

Das hier beschriebene Beispiel nutzt die Statemachine, welche in dem Kapitel „Allgemeine Server - Projektstruktur [► 628]“ beschrieben ist. Die States: 0,1,10 und 100 sind identisch zu der dort beschriebenen Statemachine. Andere States wurden für das Beispiel modifiziert oder auch neue States hinzugefügt.

Ein neuer Funktionsbaustein „FB_DirectControl“ wurde hinzugefügt. In diesem ist die Statemachine für das Controlmodel implementiert. Damit diese für ein Datenobjekt ausgeführt wird, muss im „VAR_INPUT“-Teil des Funktionsbausteins des zu steuernden Datenobjekts eine „FB_DirectControl“-Instanz wie folgt deklariert werden:

```

fbControl: FB_DirectControl := (ipCtrl:=THIS^.AddServerJob(ipEvent:=fbControl, ipCompletion:=fbControl));

```

Jede dieser Instanzen verfügt über eine „Execute“-Methode, die jeden Zyklus aufgerufen werden muss, um das Controlmodel zu überwachen. In diesem Beispiel erfolgt dies im „FB_IEDServerSession“-Funktionsbaustein. In dieser Methode befinden sich die State machine sowie die Überwachung für das Versenden von „LastApplError“-Nachrichten. Zusätzlich zu der „Execute“-Methode verfügt der „FB_DirectControl“-Funktionsbaustein über weitere Methoden und Eigenschaften, die für das Controlmodel benötigt werden. Diese werden in den folgenden Tabellen erläutert.

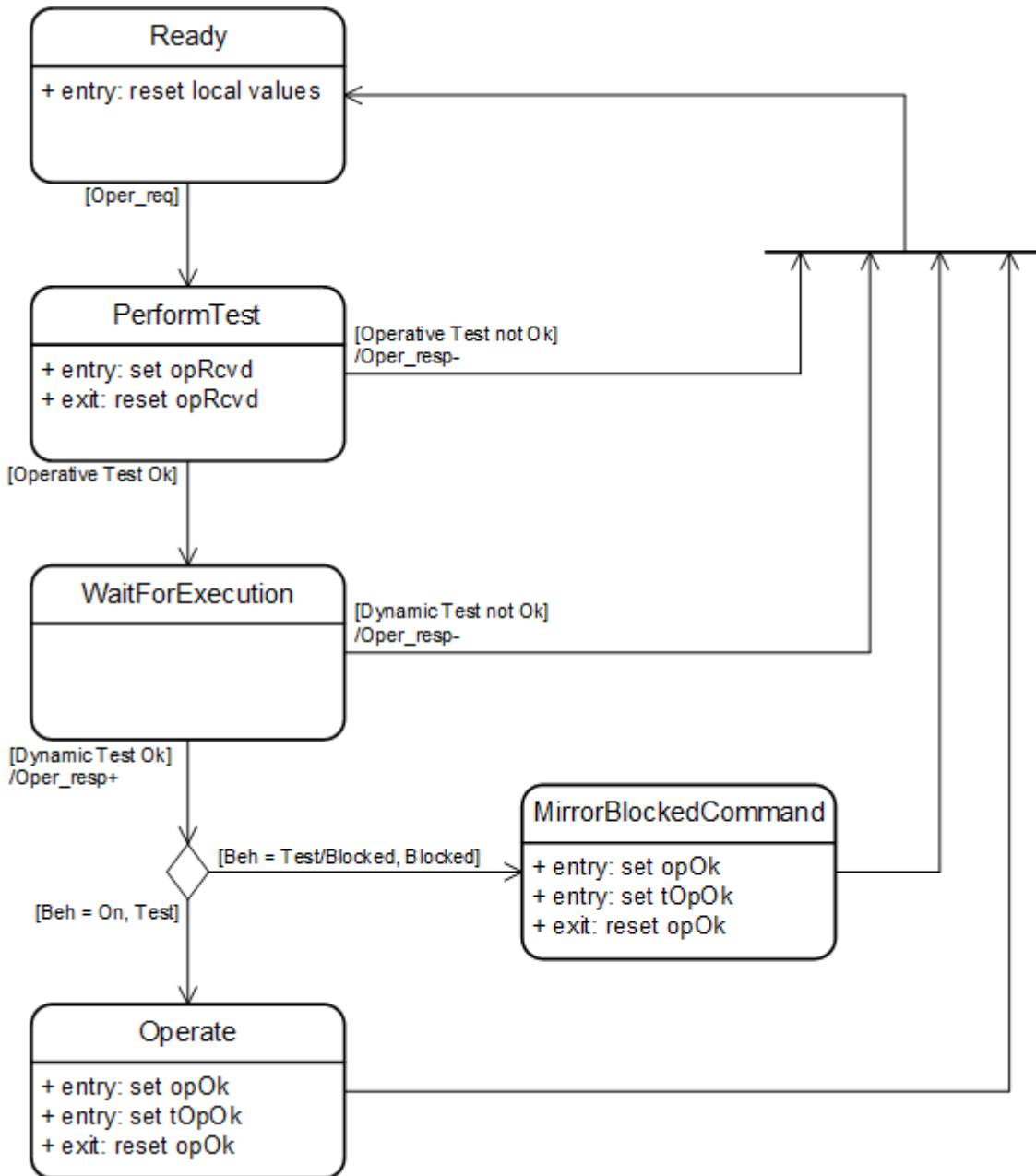
Methoden des "FB_DirectControl"-Funktionsbausteins

| Methodenname | Beschreibung |
|-----------------------|---|
| _CheckCtlVal | Entsprechend der steuerbaren CDC des Datenobjekts wird überprüft, ob der zu schreibende Wert von „ctlVal“ gültig ist. Falls der Wert ungültig ist, wird ein entsprechender Fehlercode zurückgegeben. |
| _DynamicTest | Diese Methode führt den dynamischen Test aus. Da dieser in diesem Beispiel nicht benötigt wird, wird immer ein erfolgreiches Ergebnis zurückgegeben. Ein eigener dynamischer Test kann in dieser Methode implementiert werden. |
| _Init | Initialisiert den „FB_DirectControl“-Funktionsbaustein, indem die CDC des zu kontrollierenden Datenobjekts bestimmt wird. |
| _OperativeTest | Diese Methode führt den operativen Test aus. Dabei wird überprüft, ob der Zustand des logischen Knotens, der dem zu steuernden Datenobjekt übergeordnet ist, das Steuern zulässt sowie ob der zu schreibende Wert gültig ist. Ist der Test erfolgreich, wird die Adresse des Clients zwischengespeichert, um identifizieren zu können, welcher Client das Controlmodel ausgelöst hat. Wenn der Test fehlschlägt, wird eine „LastApplError“-Nachricht an den Client versendet. |
| _Reset | Setzt die für ein DirectControl zwischengespeicherten Werte zurück. |
| _ResetOperValues | Setzt die Werte des „Oper“-Datenattributs auf die konfigurierten Initialwerte zurück. |
| _SendLastApplError | Versendet eine „LastApplError“-Nachricht an den Client. |
| _Set[CDC]Value | Für jede steuerbare CDC ist eine Methode hinzugefügt worden, da die meisten CDCs unterschiedliche Typen verwenden, z. B. „APC“ -> „REAL“ und „SPC“ -> „BOOL“. Da die CDCs unterschiedliche Typen verwenden, besitzt auch jede dieser Methoden für den Wert (z. B. ctlVal) einen anderen Inputparameter. Anhand dieses Parameters wird der Wert des zu steuernden Datenobjekts gesetzt. |
| _SetValue | Setzt den Wert des zu steuernden Datenobjekts entsprechend dem Wert von „ctlVal“. Je nach CDC des Datenobjekts wird dafür die entsprechende „_Set[CDC]Val“-Methode aufgerufen. |
| _UpdateTimestampValue | Setzt den Wert des Datenattributs „t“ des zu steuernden Datenobjekts auf die aktuelle Systemzeit. |
| Execute | In dieser Methode ist die Statemachine für das Controlmodel implementiert. Die anderen Methoden aus diesem Funktionsbaustein werden entsprechend der Statemachine in dieser Methode aufgerufen. |
| OnAbort | Wird aufgerufen, wenn die Verbindung des Servers abgebrochen wird und bricht das Senden der „LastApplError“-Nachricht ab. |
| OnCompletionWaitEvent | Nach jedem Schreibvorgang wird diese Methode von der Serverimplementierung aufgerufen. In dieser wird bestimmt, ob eine „Operate“-Anfrage empfangen wurde. Ist dies der Fall, wird der Server-Implementierung mitgeteilt, dass diese mit dem Versenden der Antwort auf die „Operate“-Anfrage warten muss, bis der DirectControl angeschlossen ist. |
| OnGetVEvent | Diese Methode ist Teil der „I_AcsiDataAccessEventClass“-Schnittstelle, wird jedoch für dieses Beispiel nicht benötigt. |
| OnGetVPreEvent | Diese Methode ist Teil der „I_AcsiDataAccessEventClass“-Schnittstelle, wird jedoch für dieses Beispiel nicht benötigt. |
| OnSetVEvent | Diese Methode ist Teil der „I_AcsiDataAccessEventClass“-Schnittstelle, wird jedoch für dieses Beispiel nicht benötigt. |
| OnSetVPreEvent | Diese Methode ist Teil der „I_AcsiDataAccessEventClass“-Schnittstelle und wird vor jedem Schreibzugriff auf das zu steuernde Datenobjekt aufgerufen. Diese Funktionalität wird genutzt, um das Überschreiben der „Oper“-Struktur, z. B. durch einen anderen Client, während eines Control-Vorgangs zu verhindern. |

Eigenschaften des "FB_DirectControl"-Funktionsbausteins

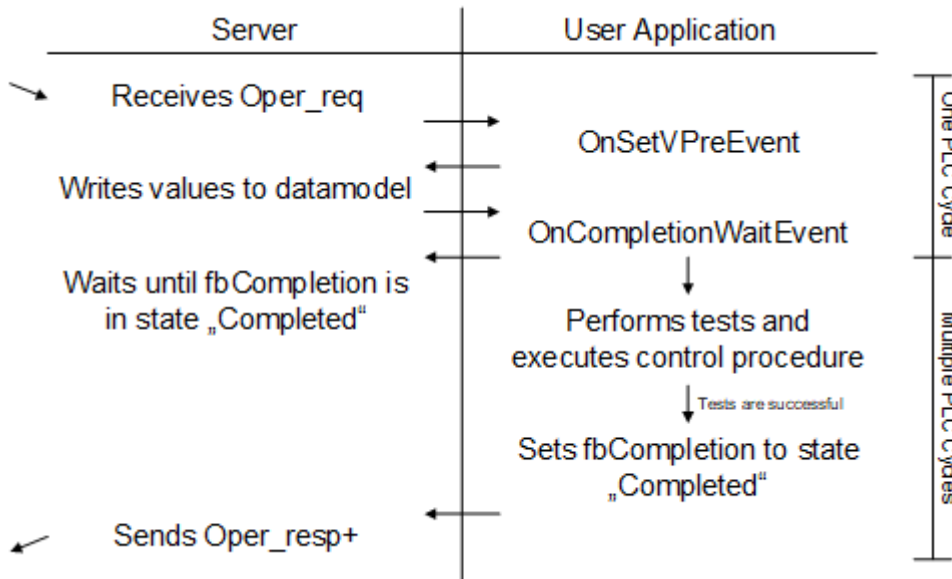
| Eigenschaftsname | Beschreibung |
|------------------|--|
| eType | Gibt die steuerbare CDC an (z. B. „APC“). |
| ipBehVal | Schnittstellenzeiger auf den Wert des „Beh“-Datenobjekts des, dem zu steuernden Datenobjekts übergeordneten, logischen Knotens. |
| ipCtrl | Schnittstellenzeiger auf das zu steuernde Datenobjekt. |
| ipStepSize | Optionaler Schnittstellenzeiger auf das „stepSize“-Datenattribut. Dieses kann für ein das Setzen des Wertes eines „BAC“-Datenobjekts genutzt werden. |

Für den Ablauf des DirectControl wird die in der unteren Abbildung dargestellte State machine genutzt, die in der „Execute“-Methode des „FB_DirectControl“-Funktionsbausteins implementiert ist. Die Übergänge zwischen den einzelnen Zuständen werden durch boolesche Variablen gesteuert. Zudem wird die boolesche Variable „bNewState“ genutzt, um neue Zustände zu erkennen und das entsprechende Verhalten beim Zustandswechsel zu implementieren.



Da zwischen Empfang der „Operate“-Anfrage und dem Ausführen mehrere SPS-Zyklen liegen können, muss der Server-Applikation mitgeteilt werden, wie lange sie bis zum Senden einer Antwort warten muss. Dafür wird eine „fbCompletion“ genannte Instanz des „FB_AcsiServiceResultClass“-Funktionsbausteins verwendet.

Beim Empfangen einer „Operate“-Anfrage wird dieser in der „OnCompletionWaitEvent“-Methode in den Zustand „Busy“ gesetzt und als Schnittstellenzeiger an die Server-Applikation übergeben. Diese wartet nun so lange, bis fbCompletion im Zustand „Completed“ ist und schickt dann die Antwort auf die „Operate“-Anfrage. Soll die „Operate“-Anfrage direkt abgelehnt werden, wird dies in der „OnSetVPreEvent“-Methode erledigt. Der Ablauf einer „Operate“-Anfrage ist in der folgenden Abbildung dargestellt:



Dies ist nur eine exemplarische Implementierung der DirectControl-Funktionalität in der TwinCAT SPS.

7.20 Server - Direct Control with enhanced security

Dieses Beispiel zeigt die Implementierung von ctlModel := 3 (Direct control with enhanced security, direct-operate) in einem Server für folgende CDCs:

- SPC (controllable single point)
- DPC (controllable double point)
- APC (controllable analogue process value)
- BAC (binary controlled analogue process value)
- BSC (binary controlled step position information)
- ENC (controllable enumerated status)
- INC (controllable integer status)
- ISC (integer controlled step position information)

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1031/TF6510_TC3_IEC61850/Resources/10698256779.zip

Das hier beschriebene Beispiel nutzt die Statemachine, welche in dem Kapitel „Allgemeine Server - Projektstruktur [▶ 628]“ beschrieben ist. Die States: 0,1,10 und 100 sind identisch zu der dort beschriebenen Statemachine. Andere States wurden für das Beispiel modifiziert oder auch neue States hinzugefügt.

Ein neuer Funktionsbaustein „FB_DirectControl“ wurde hinzugefügt. In diesem ist die Statemachine für das Controlmodel implementiert. Damit diese für ein Datenobjekt ausgeführt wird, muss im „VAR_INPUT“-Teil des Funktionsbausteins des zu steuernden Datenobjekts eine „FB_DirectControl“-Instanz wie folgt deklariert werden:

```
fbControl: FB_DirectControl := (ipCtrl:=THIS^.AddServerJob(ipEvent:=fbControl, ipCompletion:=fbControl));
```

Jede dieser Instanzen verfügt über eine „Execute“-Methode, die jeden Zyklus aufgerufen werden muss, um das Controlmodel zu überwachen. In diesem Beispiel wird dies im „FB_IEDServerSession“-Funktionsbaustein erledigt. In dieser Methode befinden sich die Statemachine sowie die Überwachung für

das Versenden von „LastApplError“ und „CommandTermination“-Nachrichten. Zusätzlich zu der „Execute“-Methode verfügt der „FB_DirectControl“-Funktionsbaustein über weitere Methoden und Eigenschaften, die für das Controlmodel benötigt werden. Diese werden in den folgenden Tabellen erläutert.

Methoden des "FB_DirectControl"-Funktionsbausteins

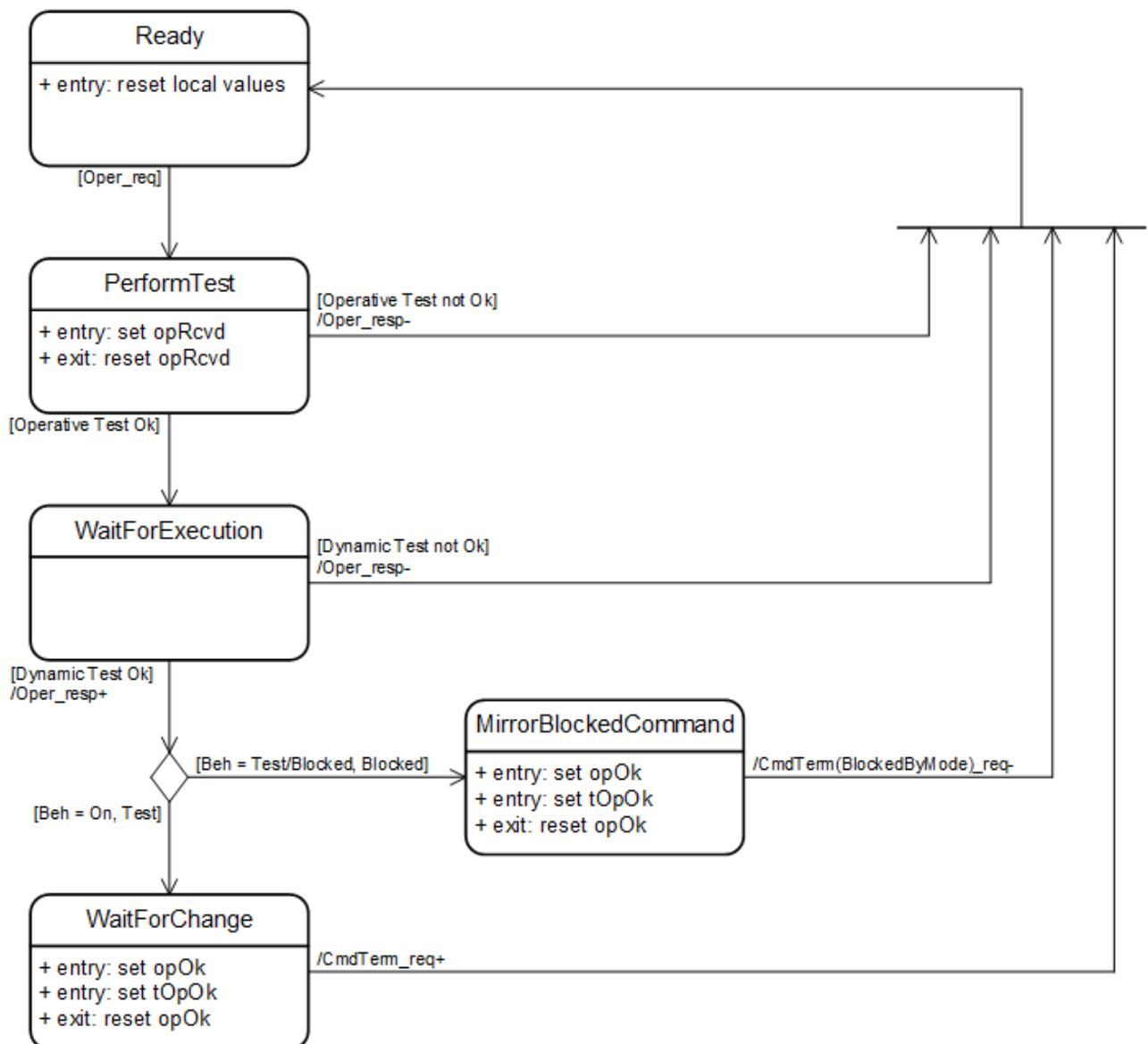
| Methodenname | Beschreibung |
|--------------------------------|--|
| _CheckCtlVal | Entsprechend der steuerbaren CDC des Datenobjekts wird überprüft, ob der zu schreibende Wert von „ctlVal“ gültig ist. Falls der Wert ungültig ist, wird ein entsprechender Fehlercode zurückgegeben. |
| _DynamicTest | Diese Methode führt den dynamischen Test aus. Da dieser in diesem Beispiel nicht benötigt wird, wird immer ein erfolgreiches Ergebnis zurückgegeben. Ein eigener dynamischer Test kann in dieser Methode implementiert werden. |
| _Init | Initialisiert den „FB_DirectControl“-Funktionsbaustein, indem die CDC des zu kontrollierenden Datenobjekts bestimmt wird. |
| _OperativeTest | Diese Methode führt den operativen Test aus. Dabei wird überprüft, ob der Zustand des logischen Knotens, der dem zu steuernden Datenobjekt übergeordnet ist, das Steuern zulässt sowie ob der zu schreibende Wert gültig ist. Ist der Test erfolgreich, wird die Adresse des Clients zwischengespeichert, um identifizieren zu können, welcher Client das Controlmodel ausgelöst hat. Wenn der Test fehlschlägt, wird eine „LastAppError“-Nachricht an den Client versendet. |
| _Reset | Setzt die für ein DirectControl zwischengespeicherten Werte zurück. |
| _ResetOperValues | Setzt die Werte des „Oper“-Datenattributs auf die konfigurierten Initialwerte zurück. |
| _SendCommandTerminationReq Neg | Versendet eine negative „CommandTermination“-Anfrage an den Client. |
| _SendCommandTerminationReq Pos | Versendet eine positive „CommandTermination“-Anfrage an den Client. |
| _SendLastAppError | Versendet eine „LastAppError“-Nachricht an den Client. |
| _Set[CDC]Value | Für jede steuerbare CDC ist eine Methode hinzugefügt worden, da die meisten CDCs unterschiedliche Typen verwenden, z. B. „APC“ -> „REAL“ und „SPC“ -> „BOOL“. Da die CDCs unterschiedliche Typen verwenden, besitzt auch jede dieser Methoden für den Wert (z. B. ctlVal) einen anderen Inputparameter. Anhand dieses Parameters wird der Wert des zu steuernden Datenobjekts gesetzt. |
| _SetValue | Setzt den Wert des zu steuernden Datenobjekts entsprechend dem Wert von „ctlVal“. Je nach CDC des Datenobjekts wird dafür die entsprechende „_Set[CDC]Val“-Methode aufgerufen. |
| _UpdateTimestampValue | Setzt den Wert des Datenattributs „t“ des zu steuernden Datenobjekts auf die aktuelle Systemzeit. |
| Execute | In dieser Methode ist die Statemachine für das Controlmodel implementiert. Die anderen Methoden aus diesem Funktionsbaustein werden entsprechend der Statemachine in dieser Methode aufgerufen. |
| OnAbort | Wird aufgerufen, wenn die Verbindung des Servers abgebrochen wird und bricht das Senden der „LastAppError“- und „CommandTermination“-Nachricht ab. |
| OnCompletionWaitEvent | Nach jedem Schreibvorgang wird diese Methode von der Serverimplementierung aufgerufen. In dieser wird bestimmt, ob eine „Operate“-Anfrage empfangen wurde. Ist dies der Fall wird der Server-Implementierung mitgeteilt, dass diese mit dem Versenden der Antwort auf die „Operate“-Anfrage warten muss, bis der DirectControl angeschlossen ist. |
| OnGetVEvent | Diese Methode ist Teil der „I_AcsiDataAccessEventClass“-Schnittstelle, wird jedoch für dieses Beispiel nicht benötigt. |
| OnGetVPreEvent | Diese Methode ist Teil der „I_AcsiDataAccessEventClass“-Schnittstelle, wird jedoch für dieses Beispiel nicht benötigt. |
| OnSetVEvent | Diese Methode ist Teil der „I_AcsiDataAccessEventClass“-Schnittstelle, wird jedoch für dieses Beispiel nicht benötigt. |

| Methodenname | Beschreibung |
|----------------|---|
| OnSetVPreEvent | Diese Methode ist Teil der „I_AcsiDataAccessEventClass“-Schnittstelle und wird vor jedem Schreibzugriff auf das zu steuernde Datenobjekt aufgerufen. Diese Funktionalität wird genutzt, um das Überschreiben der „Oper“-Struktur, z. B. durch einen anderen Client, während eines Control-Vorgangs zu verhindern. |

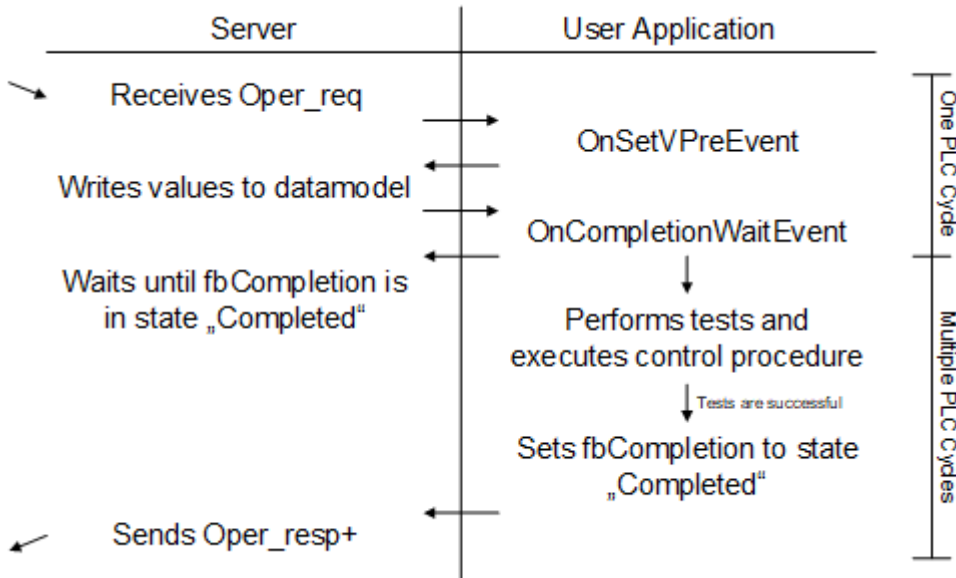
Eigenschaften des "FB_DirectControl"-Funktionsbausteins

| Eigenschaftsname | Beschreibung |
|------------------|--|
| eType | Gibt die steuerbare CDC an (z. B. „APC“). |
| ipBehVal | Schnittstellenzeiger auf den Wert des „Beh“-Datenobjekts des, dem zu steuernden Datenobjekts übergeordneten, logischen Knotens. |
| ipCtrl | Schnittstellenzeiger auf das zu steuernde Datenobjekt. |
| ipStepSize | Optionaler Schnittstellenzeiger auf das „stepSize“-Datenattribut. Dieses kann für ein das Setzen des Wertes eines „BAC“-Datenobjekts genutzt werden. |

Für den Ablauf des DirectControl wird die in der unteren Abbildung dargestellte Statemaschine genutzt, die in der „Execute“-Methode des „FB_DirectControl“-Funktionsbausteins implementiert ist. Die Übergänge zwischen den einzelnen Zuständen werden durch boolesche Variablen gesteuert. Zudem wird die boolesche Variable „bNewState“ genutzt, um neue Zustände zu erkennen und das entsprechende Verhalten beim Zustandswechsel zu implementieren.



Da zwischen Empfang der „Operate“-Anfrage und dem Ausführen mehrere SPS-Zyklen liegen können, muss der Server-Applikation mitgeteilt werden, wie lange sie bis zum Senden einer Antwort warten muss. Dafür wird eine „fbCompletion“ genannte Instanz des „FB_AcsiServiceResultClass“-Funktionsbausteins verwendet. Beim Empfangen einer „Operate“-Anfrage wird dieser in der „OnCompletionWaitEvent“-Methode in den Zustand „Busy“ gesetzt und als Schnittstellenzeiger an die Server-Applikation übergeben. Diese wartet nun so lange, bis fbCompletion“ im Zustand „Completed“ ist und schickt dann die Antwort auf die „Operate“-Anfrage. Soll die „Operate“-Anfrage direkt abgelehnt werden, wird dies in der „OnSetVPreEvent“-Methode erledigt. Der Ablauf einer „Operate“-Anfrage ist in der folgenden Abbildung dargestellt:



Dies ist nur eine exemplarische Implementierung der DirectControl-Funktionalität in der TwinCAT SPS.

7.21 Server - SBO Control with normal security

Dieses Beispiel zeigt die Implementierung von `ctlModel := 2` (SBO control with normal security, operate-once or operate-many) in einem Server für folgende CDCs:

- SPC (controllable single point)
- DPC (controllable double point)
- APC (controllable analogue process value)
- BAC (binary controlled analogue process value)
- BSC (binary controlled step position information)
- ENC (controllable enumerated status)
- INC (controllable integer status)
- ISC (integer controlled step position information)

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1031/TF6510_TC3_IEC61850/Resources/10698254091.zip

Das hier beschriebene Beispiel nutzt die State machine, welche in dem Kapitel „Allgemeine Server - Projektstruktur | 628|“ beschrieben ist. Die States: 0, 1, 10 und 100 sind identisch zu der dort beschriebenen State machine. Andere States wurden für das Beispiel modifiziert oder auch neue States hinzugefügt.

Ein neuer Funktionsbaustein „FB_SboControl“ wurde hinzugefügt. In diesem ist die State machine für das Controlmodel implementiert. Damit diese für ein Datenobjekt ausgeführt wird, muss im „VAR_INPUT“-Teil des Funktionsbausteins des zu steuernden Datenobjekts eine „FB_SboControl“-Instanz wie folgt deklariert werden:

```
fbControl: FB_SboControl := (ipCtrl:=THIS^.AddServerJob(ipEvent:=fbControl, ipCompletion:=fbControl));
```


Jede dieser Instanzen verfügt über eine „Execute“-Methode, die jeden Zyklus aufgerufen werden muss, um das Controlmodel zu überwachen. In diesem Beispiel wird dies im „FB_IEDServerSession“-Funktionsbaustein erledigt. In dieser Methode befinden sich die StateMachine sowie die Überwachung für das Versenden von „LastApplError“-Nachrichten. Zusätzlich zu der „Execute“-Methode verfügt der „FB_SboControl“-Funktionsbaustein über weitere Methoden und Eigenschaften, die für das Controlmodel benötigt werden. Diese werden in den folgenden Tabellen erläutert.

Methoden des „FB_SboControl“-Funktionsbausteins

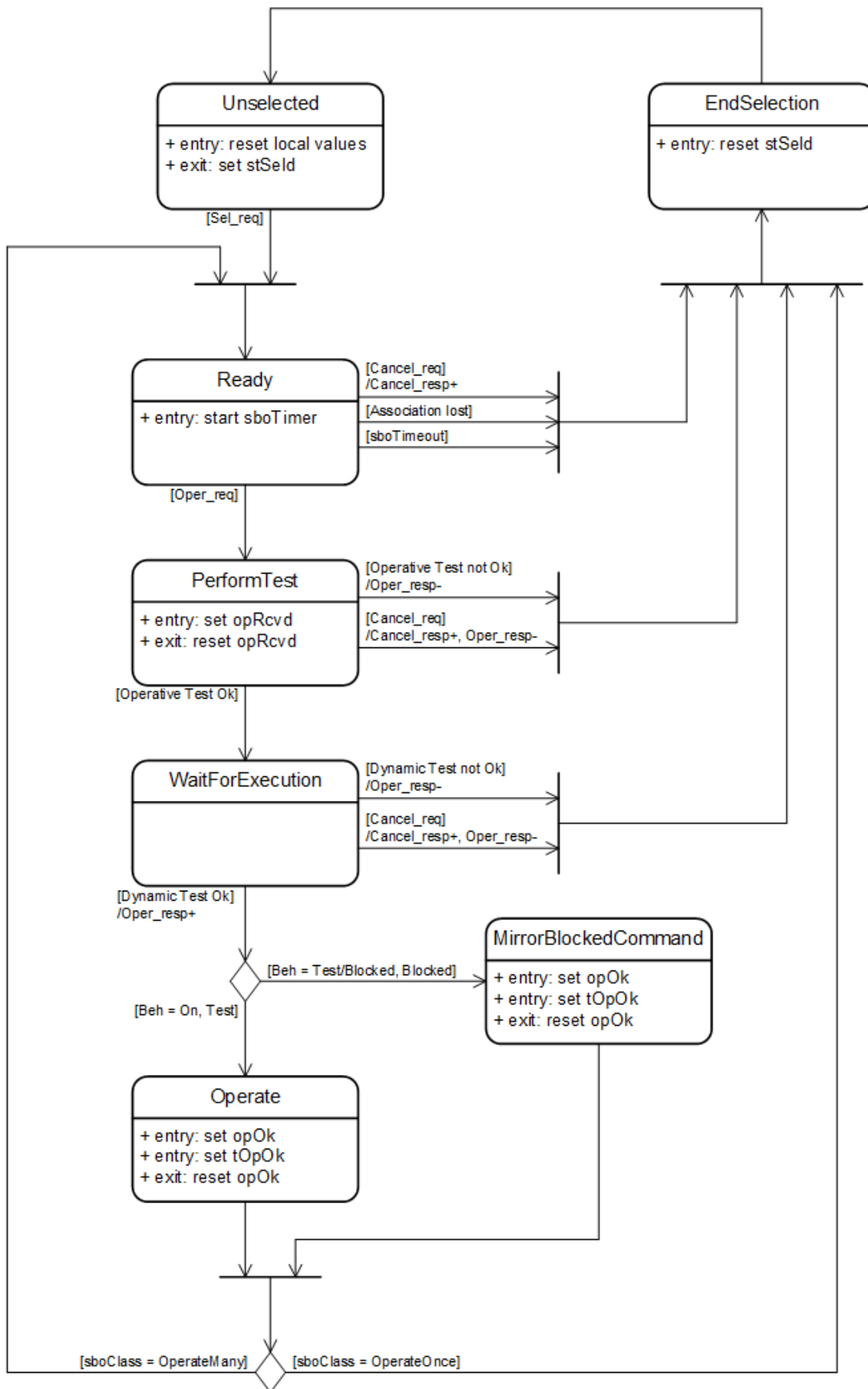
| Methodenname | Beschreibung |
|-----------------------|---|
| _CheckCancelReq | Wird beim Empfangen einer „Cancel“-Anfrage aufgerufen und prüft, ob diese gültig ist. Falls diese ungültig ist, wird eine entsprechende „LastApplError“-Nachricht versendet. |
| _CheckCtlVal | Entsprechend der steuerbaren CDC des Datenobjekts wird überprüft, ob der zu schreibende Wert von „ctlVal“ gültig ist. Falls der Wert ungültig ist, wird ein entsprechender Fehlercode zurückgegeben. |
| _DynamicTest | Diese Methode führt den dynamischen Test aus. Da dieser in diesem Beispiel nicht benötigt wird, wird immer ein erfolgreiches Ergebnis zurückgegeben. Ein eigener dynamischer Test kann in dieser Methode implementiert werden. |
| _Init | Initialisiert den „FB_DirectControl“-Funktionsbaustein, indem die CDC des zu kontrollierenden Datenobjekts bestimmt wird. |
| _OperativeTest | Diese Methode führt den operativen Test aus. Dabei wird überprüft, ob der Zustand des logischen Knotens, der dem zu steuernden Datenobjekt übergeordnet ist, das Steuern zulässt sowie ob der zu schreibende Wert gültig ist. Ist der Test erfolgreich, wird die Adresse des Clients zwischengespeichert, um identifizieren zu können, welcher Client das Controlmodel ausgelöst hat. Wenn der Test fehlschlägt, wird eine „LastApplError“-Nachricht an den Client versendet. |
| _Reset | Setzt die für ein SBOControl zwischengespeicherten Werte zurück. |
| _ResetCancelValues | Setzt die Werte des „Cancel“-Datenattributs auf die konfigurierten Initialwerte zurück. |
| _ResetOperValues | Setzt die Werte des „Oper“-Datenattributs auf die konfigurierten Initialwerte zurück. |
| _SendLastApplError | Versendet eine „LastApplError“-Nachricht an den Client. |
| _Set[CDC]Value | Für jede steuerbare CDC ist eine Methode hinzugefügt worden, da die meisten CDCs unterschiedliche Typen verwenden, z. B. „APC“ -> „REAL“ und „SPC“ -> „BOOL“. Da die CDCs unterschiedliche Typen verwenden, besitzt auch jede dieser Methoden für den Wert (z. B. ctlVal) einen anderen Inputparameter. Anhand dieses Parameters wird der Wert des zu steuernden Datenobjekts gesetzt. |
| _SetValue | Setzt den Wert des zu steuernden Datenobjekts entsprechend dem Wert von „ctlVal“. Je nach CDC des Datenobjekts wird dafür die entsprechende „_Set[CDC]Val“-Methode aufgerufen. |
| _UpdateTimestampValue | Setzt den Wert des Datenattributs „t“ des zu steuernden Datenobjekts auf die aktuelle Systemzeit. |
| Execute | In dieser Methode ist die Statemachine für das Controlmodel implementiert. Die anderen Methoden aus diesem Funktionsbaustein werden entsprechend der Statemachine in dieser Methode aufgerufen. |
| OnAbort | Wird aufgerufen, wenn die Verbindung des Servers abgebrochen wird und bricht das Senden der „LastApplError“-Nachricht ab. |
| OnCompletionWaitEvent | Nach jedem Schreibvorgang wird diese Methode von der Serverimplementierung aufgerufen. In dieser wird bestimmt, ob eine „Operate“-Anfrage empfangen wurde. Ist dies der Fall wird der Server-Implementierung mitgeteilt, dass diese mit dem Versenden der Antwort auf die „Operate“-Anfrage warten muss, bis der SBOControl angeschlossen ist. |
| OnGetVEvent | Diese Methode ist Teil der „I_AcsiDataAccessEventClass“-Schnittstelle und wird nach jedem Lesezugriff auf das zu steuernde Datenobjekt aufgerufen. Diese Funktionalität wird genutzt, um den Wert des Datenattributs „SBO“ nach einer „Select“-Anfrage zurückzusetzen. |
| OnGetVPreEvent | Diese Methode ist Teil der „I_AcsiDataAccessEventClass“-Schnittstelle und wird vor jedem Lesezugriff auf das zu steuernde Datenobjekt aufgerufen. Diese Funktionalität wird genutzt, um „Select“-Anfragen zu überprüfen und entsprechend anzunehmen oder abzulehnen. |
| OnSetVEvent | Diese Methode ist Teil der „I_AcsiDataAccessEventClass“-Schnittstelle, wird jedoch für dieses Beispiel nicht benötigt. |

| Methodenname | Beschreibung |
|----------------|--|
| OnSetVPreEvent | Diese Methode ist Teil der „I_AcsiDataAccessEventClass“-Schnittstelle und wird vor jedem Schreibzugriff auf das zu steuernde Datenobjekt aufgerufen. Diese Funktionalität wird genutzt, um das Überschreiben der „Oper“- und „Cancel“-Struktur sowie des „SBO“-Datenattributs, z. B. durch einen anderen Client, während eines Control-Vorgangs zu verhindern. |

Eigenschaften des "FB_SboControl"-Funktionsbausteins

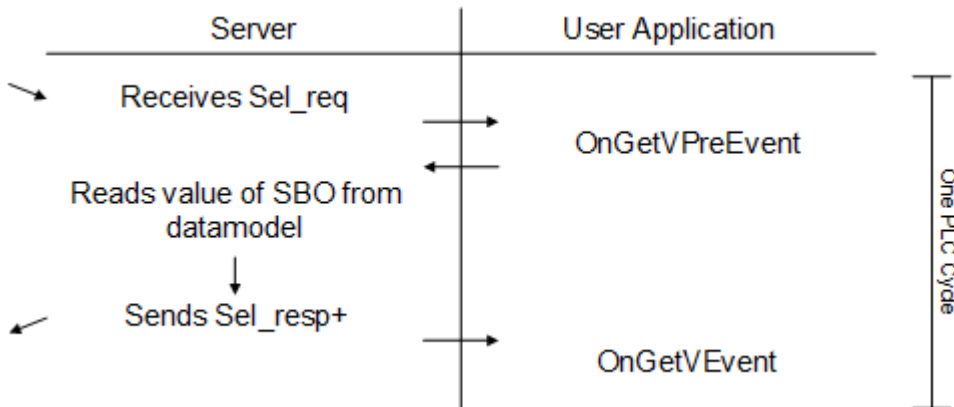
| Eigenschaftsname | Beschreibung |
|------------------|--|
| eType | Gibt die steuerbare CDC an (z. B. „APC“). |
| ipBehVal | Schnittstellenzeiger auf den Wert des „Beh“-Datenobjekts des, dem zu steuernden Datenobjekts übergeordneten, logischen Knotens. |
| ipCtrl | Schnittstellenzeiger auf das zu steuernde Datenobjekt. |
| ipStepSize | Optionaler Schnittstellenzeiger auf das „stepSize“-Datenattribut. Dieses kann für ein das Setzen des Wertes eines „BAC“-Datenobjekts genutzt werden. |

Für den Ablauf des SBOControl wird die in der unteren Abbildung dargestellte State Machine genutzt, die in der „Execute“-Methode des „FB_SboControl“-Funktionsbausteins implementiert ist. Die Übergänge zwischen den einzelnen Zuständen werden durch boolesche Variablen gesteuert. Zudem wird die boolesche Variable „bNewState“ genutzt, um neue Zustände zu erkennen und das entsprechende Verhalten beim Zustandswechsel zu implementieren.

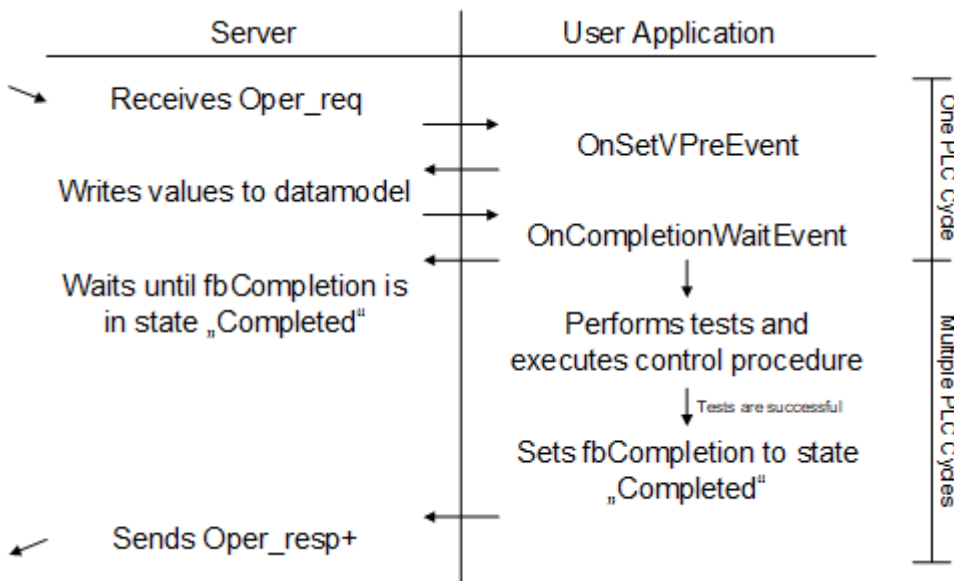


Da es sich bei einer „Select“-Anfrage um eine Leseanfrage auf das „SBO“-Datenattribut handelt, wird diese in der „OnGetVPreEvent“-Methode überprüft. In dieser Methode kann die „Select“-Anfrage angenommen werden, indem der Wert von „SBO“ auf die Referenz auf das „SBO“-Datenattribut gesetzt wird. Um die

Anfrage abzulehnen wird der Wert auf einen leeren String gesetzt. Nach dem Senden der Antwort wird die „OnGetVEvent“-Methode aufgerufen, in der der Wert des Datenattributs zurückgesetzt werden kann. Der Ablauf einer erfolgreichen „Select“-Anfrage ist in der folgenden Abbildung dargestellt.



Da zwischen Empfang der „Operate“-Anfrage und dem Ausführen mehrere SPS-Zyklen liegen können, muss der Server-Applikation mitgeteilt werden, wie lange sie bis zum Senden einer Antwort warten muss. Dafür wird eine „fbCompletion“ genannte Instanz des „FB_AcsiServiceResultClass“-Funktionsbausteins verwendet. Beim Empfangen einer „Operate“-Anfrage wird dieser in der „OnCompletionWaitEvent“-Methode in den Zustand „Busy“ gesetzt und als Schnittstellenzeiger an die Server-Applikation übergeben. Diese wartet nun so lange, bis „fbCompletion“ im Zustand „Completed“ ist und schickt dann die Antwort auf die „Operate“-Anfrage. Soll die „Operate“-Anfrage direkt abgelehnt werden, wird dies in der „OnSetVPreEvent“-Methode erledigt. Der Ablauf einer „Operate“-Anfrage ist in der folgenden Abbildung dargestellt:



Dies ist nur eine exemplarische Implementierung der SBOControl-Funktionalität in der TwinCAT SPS.

7.22 Server - SBO Control with enhanced security

Dieses Beispiel zeigt die Implementierung von `ctlModel := 4` (SBO control with enhanced security, operate-once or operate-many) in einem Server für folgende CDCs:

- SPC (controllable single point)
- DPC (controllable double point)
- APC (controllable analogue process value)
- BAC (binary controlled analogue process value)
- BSC (binary controlled step position information)

- ENC (controllable enumerated status)
- INC (controllable integer status)
- ISC (integer controlled step position information)

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1031/TF6510_TC3_IEC61850/Resources/10698259467.zip

Das hier beschriebene Beispiel nutzt die State Machine, welche in dem Kapitel „Allgemeine Server - Projektstruktur [▶ 628]“ beschrieben ist. Die States: 0,1,10 und 100 sind identisch zu der dort beschriebenen State Machine. Andere States wurden für das Beispiel modifiziert oder auch neue States hinzugefügt.

Ein neuer Funktionsbaustein „FB_SboControl“ wurde hinzugefügt. In diesem ist die State Machine für das Controlmodel implementiert. Damit diese für ein Datenobjekt ausgeführt wird, muss im „VAR_INPUT“-Teil des Funktionsbausteins des zu steuernden Datenobjekts eine „FB_SboControl“-Instanz wie folgt deklariert werden:

```
fbControl: FB_SboControl := (ipCtrl:=THIS^.AddServerJob(ipEvent:=fbControl, ipCompletion:=fbControl));
```

Jede dieser Instanzen verfügt über eine „Execute“-Methode, die jeden Zyklus aufgerufen werden muss, um das Controlmodel zu überwachen. In diesem Beispiel wird dies im „FB_IEDServerSession“-Funktionsbaustein erledigt. In dieser Methode befinden sich die State Machine sowie die Überwachung für das Versenden von „LastApplError“ und „CommandTermination“-Nachrichten. Zusätzlich zu der „Execute“-Methode verfügt der „FB_SboControl“-Funktionsbaustein über weitere Methoden und Eigenschaften, die für das Controlmodel benötigt werden. Diese werden in den folgenden Tabellen erläutert.

Methoden des „FB_SboControl“-Funktionsbausteins

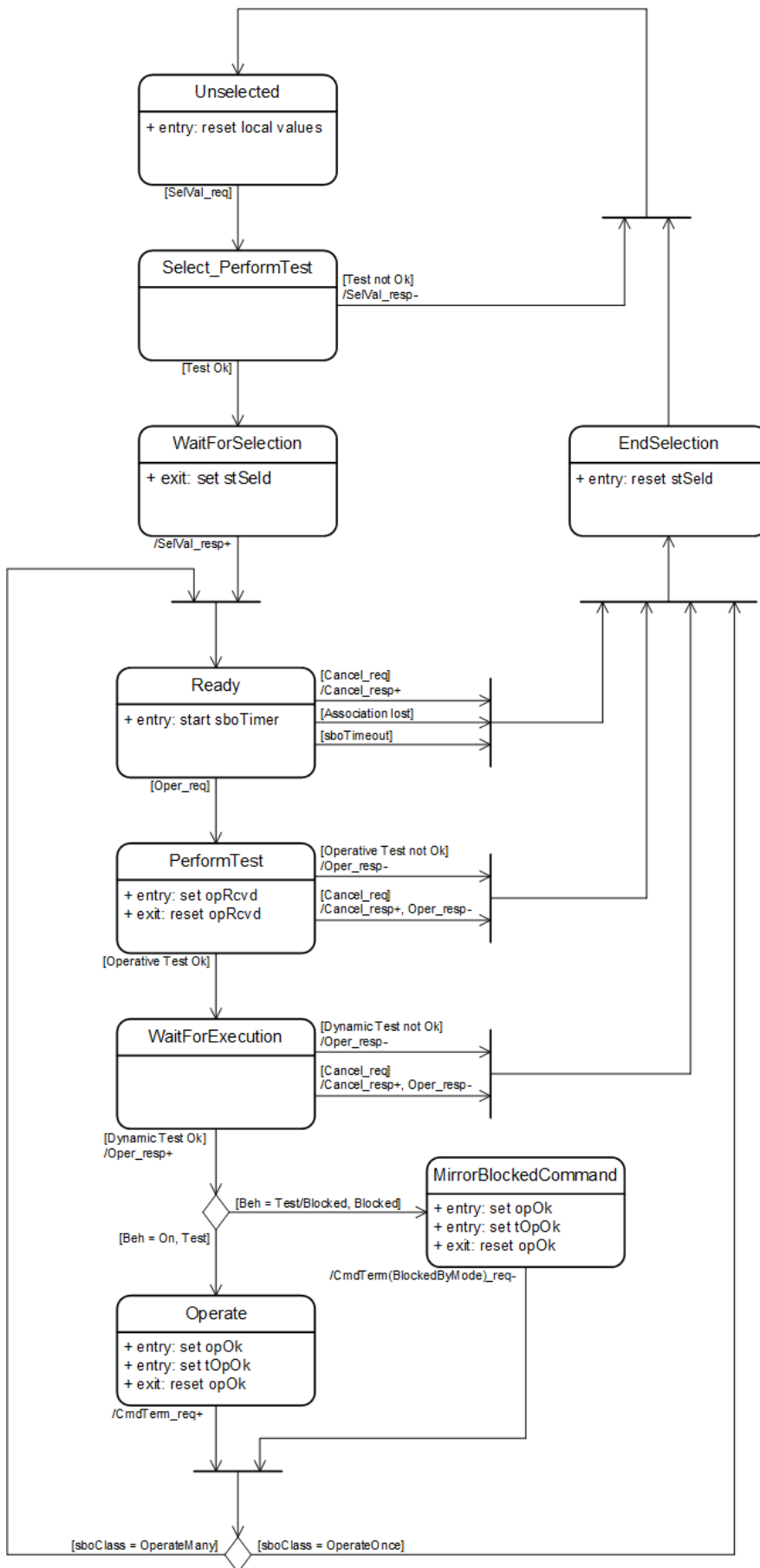
| Methodenname | Beschreibung |
|-----------------------------------|---|
| _CompareCtlVal | Überprüft entsprechend der steuerbaren CDC des zu steuernden Datenobjekts ob zwei Werte von „ctlVal“ gleich sind. Diese Methode wird genutzt um zu prüfen, ob die „SelectWithValue“- und „Operate“-Anfrage identische Werte enthalten. |
| _CheckCancelReq | Wird beim Empfangen einer „Cancel“-Anfrage aufgerufen und prüft, ob diese gültig ist. Falls diese ungültig ist, wird eine entsprechende „LastApplError“-Nachricht versendet. |
| _CheckCtlVal | Entsprechend der steuerbaren CDC des Datenobjekts wird überprüft, ob der zu schreibende Wert von „ctlVal“ gültig ist. Falls der Wert ungültig ist, wird ein entsprechender Fehlercode zurückgegeben. |
| _DynamicTest | Diese Methode führt den dynamischen Test aus. Da dieser in diesem Beispiel nicht benötigt wird, wird immer ein erfolgreiches Ergebnis zurückgegeben. Ein eigener dynamischer Test kann in dieser Methode implementiert werden. |
| _Init | Initialisiert den „FB_DirectControl“-Funktionsbaustein, indem die CDC des zu kontrollierenden Datenobjekts bestimmt wird. |
| _OperativeTest | Diese Methode führt den operativen Test aus. Dabei wird überprüft, ob der Zustand des logischen Knotens, der dem zu steuernden Datenobjekt übergeordnet ist, das Steuern zulässt sowie ob der zu schreibende Wert gültig ist. Ist der Test erfolgreich, wird die Adresse des Clients zwischengespeichert, um identifizieren zu können, welcher Client das Controlmodel ausgelöst hat. Wenn der Test fehlschlägt, wird eine „LastApplError“-Nachricht an den Client versendet. |
| _Reset | Setzt die für ein SBOControl zwischengespeicherten Werte zurück. |
| _ResetCancelValues | Setzt die Werte des „Cancel“-Datenattributs auf die konfigurierten Initialwerte zurück. |
| _ResetSBOwValues | Setzt die Werte des „SBOw“-Datenattributs auf die konfigurierten Initialwerte zurück. |
| _ResetOperValues | Setzt die Werte des „Oper“-Datenattributs auf die konfigurierten Initialwerte zurück. |
| _SendCommandTerminationReq
Neg | Versendet eine negative „CommandTermination“-Anfrage an den Client. |
| _SendCommandTerminationReq
Pos | Versendet eine positive „CommandTermination“-Anfrage an den Client. |
| _SendLastApplError | Versendet eine „LastApplError“-Nachricht an den Client. |
| _Set[CDC]Value | Für jede steuerbare CDC ist eine Methode hinzugefügt worden, da die meisten CDCs unterschiedliche Typen verwenden, z. B. „APC“ -> „REAL“ und „SPC“ -> „BOOL“. Da die CDCs unterschiedliche Typen verwenden, besitzt auch jede dieser Methoden für den Wert (z. B. ctlVal) einen anderen Inputparameter. Anhand dieses Parameters wird der Wert des zu steuernden Datenobjekts gesetzt. |
| _SetValue | Setzt den Wert des zu steuernden Datenobjekts entsprechend dem Wert von „ctlVal“. Je nach CDC des Datenobjekts wird dafür die entsprechende „_Set[CDC]Val“-Methode aufgerufen. |
| _Test | Diese Methode führt Test der in der „SelectWithValue“-Anfrage übermittelten Daten aus. Dabei wird überprüft, ob der Zustand des logischen Knotens, der dem zu steuernden Datenobjekt übergeordnet ist, das Steuern zulässt sowie ob der zu schreibende Wert gültig ist. Ist der Test erfolgreich, wird die Adresse des Clients zwischengespeichert, um identifizieren zu können, welcher Client das Controlmodel ausgelöst hat. Wenn der Test fehlschlägt, wird eine „LastApplError“-Nachricht an den Client versendet. |
| _UpdateTimestampValue | Setzt den Wert des Datenattributs „t“ des zu steuernden Datenobjekts auf die aktuelle Systemzeit. |
| Execute | In dieser Methode ist die Statemachine für das Controlmodel implementiert. Die anderen Methoden aus diesem Funktionsbaustein werden entsprechend der Statemachine in dieser Methode aufgerufen. |

| Methodenname | Beschreibung |
|-----------------------|---|
| OnAbort | Wird aufgerufen, wenn die Verbindung des Servers abgebrochen wird und bricht das Senden der „LastAppLError“-Nachricht ab. |
| OnCompletionWaitEvent | Nach jedem Schreibvorgang wird diese Methode von der Serverimplementierung aufgerufen. In dieser wird bestimmt, ob eine „Operate“- oder „SelectWithValue“-Anfrage empfangen wurde. Ist dies der Fall wird der Server-Implementierung mitgeteilt, dass diese mit dem Versenden der Antwort auf die „Operate“- oder „SelectWithValue“-Anfrage warten muss bis der SBOControl angeschlossen ist. |
| OnGetVEvent | Diese Methode ist Teil der „I_AcsiDataAccessEventClass“-Schnittstelle, wird jedoch für dieses Beispiel nicht benötigt. |
| OnGetVPreEvent | Diese Methode ist Teil der „I_AcsiDataAccessEventClass“-Schnittstelle, wird jedoch für dieses Beispiel nicht benötigt. |
| OnSetVEvent | Diese Methode ist Teil der „I_AcsiDataAccessEventClass“-Schnittstelle, wird jedoch für dieses Beispiel nicht benötigt. |
| OnSetVPreEvent | Diese Methode ist Teil der „I_AcsiDataAccessEventClass“-Schnittstelle und wird vor jedem Schreibzugriff auf das zu steuernde Datenobjekt aufgerufen. Diese Funktionalität wird genutzt, um das Überschreiben der „Oper“-, „SBOw“- und „Cancel“-Struktur, z. B. durch einen anderen Client, während eines Control-Vorgangs zu verhindern. |

Eigenschaften des "FB_SboControl"-Funktionsbausteins

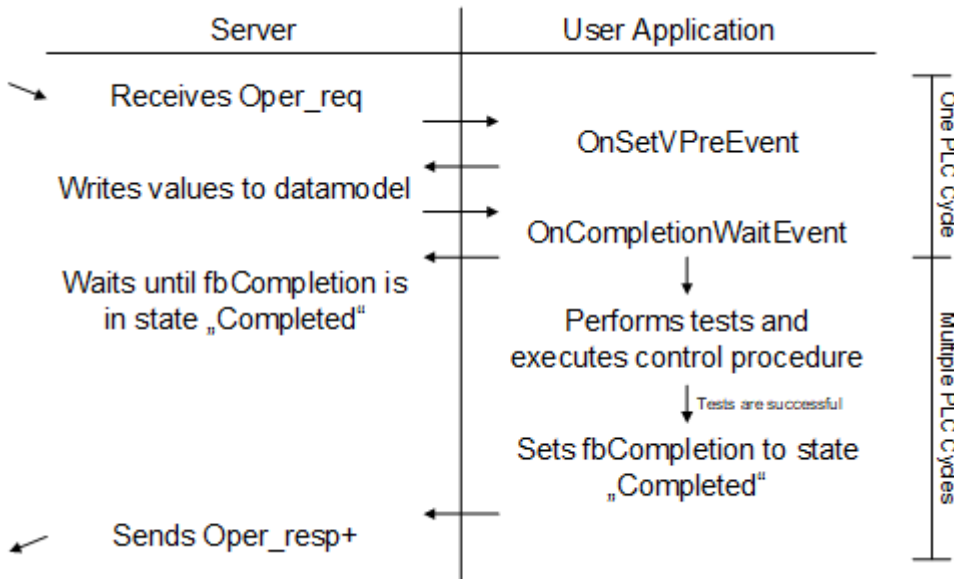
| Eigenschaftsname | Beschreibung |
|------------------|--|
| eType | Gibt die steuerbare CDC an (z. B. „APC“). |
| ipBehVal | Schnittstellenzeiger auf den Wert des „Beh“-Datenobjekts des, dem zu steuernden Datenobjekts übergeordneten, logischen Knotens. |
| ipCtrl | Schnittstellenzeiger auf das zu steuernde Datenobjekt. |
| ipStepSize | Optionaler Schnittstellenzeiger auf das „stepSize“-Datenattribut. Dieses kann für ein das Setzen des Wertes eines „BAC“-Datenobjekts genutzt werden. |

Für den Ablauf des SBOControl wird die in der unteren Abbildung dargestellte State Machine genutzt, die in der „Execute“-Methode des „FB_SboControl“-Funktionsbausteins implementiert ist. Die Übergänge zwischen den einzelnen Zuständen werden durch boolesche Variablen gesteuert. Zudem wird die boolesche Variable „bNewState“ genutzt, um neue Zustände zu erkennen und das entsprechende Verhalten beim Zustandswechsel zu implementieren.



Da zwischen Empfang der „Operate“-Anfrage und dem Ausführen mehrere SPS-Zyklen liegen können, muss der Server-Applikation mitgeteilt werden, wie lange sie bis zum Senden einer Antwort warten muss. Dafür wird eine „fbCompletion“ genannte Instanz des „FB_AcsiServiceResultClass“-Funktionsbausteins verwendet.

Beim Empfangen einer „Operate“-Anfrage wird dieser in der „OnCompletionWaitEvent“-Methode in den Zustand „Busy“ gesetzt und als Schnittstellenzeiger an die Server-Applikation übergeben. Diese wartet nun so lange, bis fbCompletion im Zustand „Completed“ ist und schickt dann die Antwort auf die „Operate“-Anfrage. Soll die „Operate“-Anfrage direkt abgelehnt werden, wird dies in der „OnSetVPreEvent“-Methode erledigt. Der Ablauf einer „Operate“-Anfrage ist in der folgenden Abbildung dargestellt:



Da es sich bei der „SelectWithValue“-Anfrage um eine Schreibanfrage handelt, wird der gleiche, im vorherigen Absatz beschriebene Ablauf wie für „Operate“-Anfragen genutzt, um diese zu bearbeiten.



Dies ist nur eine exemplarische Implementierung der SBOControl-Funktionalität in der TwinCAT SPS.

7.23 Server - User defined timestamp clock source

Dieses Beispiel zeigt die Implementierung einer benutzerdefinierten Uhr/Uhrzeit für Zeitstempelungsaufgaben in einem TwinCAT IEC 61850 Server Projekt. Die spontan vom Server benötigte Uhrzeit kann von einer externen Quelle (z.B. von einer GPS-Uhr) vorgegeben werden. Eine aktuelle Uhrzeit wird unter anderem beim Versenden der Reports benötigt.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1031/TF6510_TC3_IEC61850/Resources/10698262155.zip

Das hier beschriebene Beispiel nutzt die State Machine, welche in dem Kapitel „Allgemeine Server - Projektstruktur |> 628|“ beschrieben ist. Die States: 0, 1, 10 und 100 sind identisch zu der dort beschriebenen State Machine. Andere States wurden für das Beispiel modifiziert oder auch neue States hinzugefügt.

Die TwinCAT IEC 61850 Server-Implementierung verwendet die Schnittstelle:

„I_ScsmSystemClockEventSink“ um die Uhrzeit für Zeitstempelungsaufgaben abzufragen. Es ist möglich einen eigenen Funktionsbaustein zu verwenden, der diese spezielle Schnittstelle implementiert und den Server-Baustein so konfiguriert, dass dieser die neue Zeitquelle verwendet. In der globalen Variablenliste „TcTelecontrol“ wird dem Server-Baustein durch das Setzen der Eigenschaft „ipSystemClock:=fbMyClock“ die neue Zeitquelle zugewiesen. Der Server-Baustein wird danach jedes Mal die Methode „OnGetSystemTime“ aufrufen, wenn von ihm ein neuer Zeitstempel benötigt wird. In unserem Beispiel ist es dann z.B. der Fall, wenn Reporting aktiv ist und ein neuer Report gesendet werden soll. Zur Demonstrationszwecken wird die externe Zeitquelle auch abgefragt, wenn das Attribut: "IEDLD1/MMXU1.TotW" oder "IEDLD1/MMXU1.TotW.t" vom Client gelesen wird. Immer dann, wenn Client die Daten liest, wird die Methode: „OnGetVPreEvent“ des Bausteins: „FB_DO_IED_LD1_MMXU1_TotW“ aufgerufen. In der Implementierung der Methode wird über den Schnittstellenzeiger: „ipAssociation“ die Methode: „GetSystemTime(tT=>tT)“ aufgerufen. Diese Uhrzeitabfrage wird an unsere benutzerdefinierte Uhr weitergeleitet.

Die Client-Server-Kommunikation verwendet Zeitstempel in mindestens zwei verschiedenen Formaten: UTC-Time und Binary-Time. Aus diesem Grund liefert diese Methode den Zeitstempel als Ausgangsvariablen in diesen zwei Formaten.

```
VAR_GLOBAL
    ipCreator: I_AcsiCodeCreatorClass := GVL_AcsiVars.Creator.SetCodeRev(codeRev:=3).SetGuiVer(major:=1, minor:=1, build:=94, revision:=4);
    fbIED: FB_IED_IED;
    fbMyClock: FB_MyClock;
    fbIEDServer: FB_iec61850ServerClass := (ipIED:=fbIED, ipSystemClock:=fbMyClock, settings:=(bEnable:=TRUE, sLocalHost:='127.0.0.1'));
    fbIEDServerSession1: FB_IEDServerSession := (fbConnection:=(ipServer:=fbIEDServer, settings:=(bEnable:=TRUE)));
END_VAR
```

Das Beispiel demonstriert eine sehr einfache Software-Uhr, die mit Hilfe des RTC_EX-Funktionsbausteins realisiert wurde.

```
FUNCTION_BLOCK FB_MyClock IMPLEMENTS I_ScsmSystemClockEventSink
VAR
    _tT : T_UtcTime:=(secondSinceEpoch:=DT#2021-04-01-00:00:00, quality:=(ClockNotSynchronized:=TRUE, ClockFailure:=FALSE, LeapSecondsKnown:=FALSE), fractionOfSecond:=[0,0,0]);(* Actual UTC time. *)
    _tB : T_BinaryTime;(* Actual binary-time (EntryTime) *)
    clock : RTC_EX:=(EN:=TRUE, PDT:=DT#2021-04-01-00:00:00, PMSEK:=0);
    refreshTimer : TON:=(IN:=TRUE, PT:=T#1S);
END_VAR

METHOD FINAL Execute : BOOL
VAR_INPUT
END_VAR

refreshTimer();
IF Execute:=refreshTimer.Q THEN
    refreshTimer(IN:=FALSE); refreshTimer(IN:=TRUE);
    Update();
END_IF

METHOD FINAL OnGetSystemTime : BOOL
VAR_INPUT
    ipAA: I_ScsmAssociationClass;(* Application association. If = 0 => optional or unknown. *)
END_VAR
VAR_OUTPUT
    tT: T_UtcTime;(* UTC-time. *)
    tB: T_BinaryTime;(* Binary-time *)
END_VAR

tT:=_tT;
tB:=_tB;
OnGetSystemTime:=TRUE;

METHOD FINAL SetClock
VAR_INPUT
    tSet : DT;(* New time to set *)
END_VAR

clock(EN:=FALSE);
clock(EN:=TRUE, PDT:=tSet);
_tT.quality.ClockNotSynchronized:=FALSE;
Update();

METHOD FINAL Update
VAR_INPUT
END_VAR

clock();(* update clock time *)

(* convert to utc-time format *)
_tT.secondsSinceEpoch:=clock.CDT;
_tT.fractionOfSecond:=LTIME_TO_UtcTimeFractionOfSecond(in:=TIME_TO_LTIME(DWORD_TO_TIME(clock.CMSEK)));
Accuracy_To_UtcTimeQualityAccuracy(in:=E_UtcTimeAccuracy._03, bAccuracy0=>_tT.quality.Accuracy0, bAccuracy1=>_tT.quality.Accuracy1, bAccuracy2=>_tT.quality.Accuracy2, bAccuracy3=>_tT.quality.Accuracy3, bAccuracy4=>_tT.quality.Accuracy4);

(* convert to binary-time format *)
_tB.day:=Date_To_BinaryTime6Day(in:=DT_TO_DATE(clock.CDT));
_tB.timeOfDay:=DT_TO_TOD(clock.CDT) + DWORD_TO_TIME(clock.CMSEK);
```

7.24 Server - Unbuffered Reporting

Dieses Beispiel zeigt die Unbuffered-Reporting-Implementierung in einem TwinCAT IEC61850 Serverprojekt. Die fürs Reporting benötigten UrCBs (unbuffered report control blocks) und Datasets können im TwinCAT Telecontrol Configurator konfiguriert oder aus einer vorhandenen SCL-Datei (z. B. ICD-Datei) importiert werden. Die Instanzen der UrCBs und der Datasets werden während der Codegenerierung automatisch angelegt (normalerweise unterhalb von LLN0). Die Initialwerte der UrCB-Attribute können bereits im TwinCAT Telecontrol Configurator konfiguriert werden.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1031/TF6510_TC3_IEC61850/Resources/10698264843.zip

Das hier beschriebene Beispiel nutzt die Statemachine, welche in dem Kapitel „Allgemeine Server - Projektstruktur [▶ 628]“ beschrieben ist. Die States: 0,1,10 und 100 sind identisch zu der dort beschriebenen Statemachine. Andere States wurden für das Beispiel modifiziert oder auch neue States hinzugefügt.

Bei der Implementierung der Unbuffered-Reporting-Funktionalität in einem TwinCAT IEC 61850 Serverprojekt handelt es sich praktisch um eine „Blackbox“. D.h. die komplette Funktionalität ist in einem Baustein bereits gekapselt und die Applikation muss lediglich diesen Baustein aufrufen. Jeder UrCB besitzt einen solchen Baustein-Unterelement mit dem Namen: „Server“. Die SPS-Applikation muss nur die Methode: „Execute“ an dem „Server“-Baustein zyklisch aufrufen. Diese Methode ist für das Versenden der Reports und die Aktualisierung der Statusinformationen in dem dazugehörigen UrCB verantwortlich (siehe Code unten).

Die Konfiguration der Reports kann entweder auf der Serverseite direkt am UrCB oder seitens Clients via Client-Server-Dienste (GetUrCBDataValues und SetUrCBDataValues) vorgenommen werden. Die via Reporting zu kommunizierenden IED-Daten werden mit Hilfe der Datasets festgelegt. In einem Dataset können mehrere Datenreferenzen (Dataset-Member) konfiguriert werden.

Das Beispiel enthält eine einfache Simulation einiger Dataset-Memberwerte. Wenn die Variable „bSimulation“ auf TRUE gesetzt ist, dann werden die Werte alle 2 Sekunden modifiziert. Wie erkennt die UrCB-Server-Implementierung, dass ein Wert modifiziert wurde und ein Report generiert werden soll? Dies geschieht durch den Schreibzugriff auf den Attributwert (z. B.: bValue-, tValue-, iValue-, eValue-, nValue-Eigenschaft). Es wird aber nicht sofort bei jedem einzelnen Schreibzugriff auf einen Attributwert ein neuer Report generiert. Dies wäre unerwünscht. Die „Server“-Implementierung markiert intern die geänderten Dataset-Member als modifiziert. Erst beim Aufruf der „Execute“-Methode wird ein Report generiert und versendet.

Beachten Sie, dass beim Schreiben oder Forcen der Werte im TwinCAT Online-Mode keine Reports generiert werden. Die „Server“-Implementierung kann auf diese Weise nicht erkennen, wenn ein Wert modifiziert wurde. Integrity-Reports und General-Interrogation-Reports werden automatisch entsprechend der Konfiguration am UrCB generiert und versendet. Das Versenden dieser Reports muss in der SPS-Applikation nicht auf eine besondere Weise getriggert werden. Das Versenden dieser Reports übernimmt ebenfalls der zyklische Aufruf der Server-„Execute“-Methode.

Das zeitliche Verhalten der Report-Meldungen kann unabhängig für jede UrCB-Server-Instanz über zwei Eigenschaften „bReplaceBufferedMX“ und „bOverwriteSameCycleChanges“ konfiguriert werden. Weitere Informationen zur Funktionsweise dieser Eigenschaften finden Sie in der Beschreibung des Beispiels: [Server-Buffered Reporting \[▶ 662\]](#).

```
PROGRAM MAIN
VAR
  bSimulation           : BOOL:=TRUE; (* Enables/disables simulation of data update/
change *)
  tSimulation           : TIME:=T#2S; (* Cycle time of simulated data update/change *)
  fbTimer               : TON;
  tT                    : T_UtcTime;

  bReplaceBufferedMX    : BOOL; (* Toggles bReplaceBufferedMX property of urcb201 *)
  bOverwriteSameCycleChanges : BOOL; (* Toggles bOverwriteSameCycleChanges property of urcb301
*)
END_VAR

P_IEC61850MAIN();

fbIED.IEDLD1.LLN0.urcb201.Server.bReplaceBufferedMX:= bReplaceBufferedMX;
fbIED.IEDLD1.LLN0.urcb301.Server.bOverwriteSameCycleChanges:= bOverwriteSameCycleChanges;

fbTimer(IN:=bSimulation, PT:=tSimulation);
```

```

IF fbTimer.Q THEN(* Simulate server data update *)
  fbTimer(IN:=FALSE);
  fbTimer(IN:=bSimulation);

  fbIEDServer.GetSystemTime(ipAA:=0, tT=>tT);

  (* Simulate "IEDLD1/LLN0.DS1" member value change *)
  fbIED.IEDLD1.LEDGGIO1.SPCS01.t.tValue:= tT;
  fbIED.IEDLD1.LEDGGIO1.SPCS01.stVal.bValue:= NOT fbIED.IEDLD1.LEDGGIO1.SPCS01.stVal.bValue;
  fbIED.IEDLD1.LEDGGIO2.SPCS01.t.tValue:= tT;
  fbIED.IEDLD1.LEDGGIO2.SPCS01.stVal.bValue:= NOT fbIED.IEDLD1.LEDGGIO2.SPCS01.stVal.bValue;
  fbIED.IEDLD1.LEDGGIO3.SPCS01.t.tValue:= tT;
  fbIED.IEDLD1.LEDGGIO3.SPCS01.stVal.bValue:= NOT fbIED.IEDLD1.LEDGGIO3.SPCS01.stVal.bValue;
  fbIED.IEDLD1.LEDGGIO4.SPCS01.t.tValue:= tT;
  fbIED.IEDLD1.LEDGGIO4.SPCS01.stVal.bValue:= NOT fbIED.IEDLD1.LEDGGIO4.SPCS01.stVal.bValue;
  fbIED.IEDLD1.LEDGGIO5.SPCS01.t.tValue:= tT;
  fbIED.IEDLD1.LEDGGIO5.SPCS01.stVal.bValue:= NOT fbIED.IEDLD1.LEDGGIO5.SPCS01.stVal.bValue;
  fbIED.IEDLD1.LEDGGIO6.SPCS01.t.tValue:= tT;
  fbIED.IEDLD1.LEDGGIO6.SPCS01.stVal.bValue:= NOT fbIED.IEDLD1.LEDGGIO6.SPCS01.stVal.bValue;
  fbIED.IEDLD1.LEDGGIO7.SPCS01.t.tValue:= tT;
  fbIED.IEDLD1.LEDGGIO7.SPCS01.stVal.bValue:= NOT fbIED.IEDLD1.LEDGGIO7.SPCS01.stVal.bValue;
  fbIED.IEDLD1.LEDGGIO8.SPCS01.t.tValue:= tT;
  fbIED.IEDLD1.LEDGGIO8.SPCS01.stVal.bValue:= NOT fbIED.IEDLD1.LEDGGIO8.SPCS01.stVal.bValue;

  (* Simulate "IEDLD1/LLN0.DS2" member value change *)
  fbIED.IEDLD1.MMXU1.TotW.t.tValue:= tT;
  fbIED.IEDLD1.MMXU1.TotW.mag.f.fValue:= fbIED.IEDLD1.MMXU1.TotW.mag.f.fValue + 0.1;

  (* Simulate "IEDLD1/LLN0.DS3" member value change *)
  fbIED.IEDLD1.XCBR1.Pos.t.tValue:= tT;
  IF fbIED.IEDLD1.XCBR1.Pos.stVal.eValue = E_AcsiDbpos.On THEN
    fbIED.IEDLD1.XCBR1.Pos.stVal.eValue:= E_AcsiDbpos.Off;
  ELSE
    fbIED.IEDLD1.XCBR1.Pos.stVal.eValue:= E_AcsiDbpos.On;
  END IF
  fbIED.IEDLD1.XCBR1.Pos.q.OldData:= NOT fbIED.IEDLD1.XCBR1.Pos.q.OldData;
  fbIED.IEDLD1.XCBR1.Pos.q.eValidity:= SEL(fbIED.IEDLD1.XCBR1.Pos.q.OldData, E_AcsiQualityValidity
.Good, E_AcsiQualityValidity.Questionable);
END_IF

FUNCTION_BLOCK FB_IEDServerSession IMPLEMENTS I_ScsmAbortIndEventSink, I_ScsmAssociateIndEventSink,
I_ScsmReleaseIndEventSink
VAR_INPUT
  fbConnection      : FB_iec61850ConnectionClass := (ipAbortInd:=THIS^, ipAssociateInd:=THIS^, ipRe
leaseInd:=THIS^);
END_VAR
VAR
  _bAbort           : BOOL;
  _bDisconnect      : BOOL;
  _state            : BYTE;
  eState            : E_AsyncEnvironmentState;
  bBusy             : BOOL;
  bSuccess          : BOOL;
  ipResult          : I_AsyncServiceResultClass;
  sLastErrorResult  : T_MaxString;
  fbAbortReason     : FB_ServiceErrorClass := (stError:=SUCCESS_EVENT);
  sLastAbortReason  : T_MaxString;
  bSimulation       : BOOL:=TRUE;
  tSimulation       : TIME:=T#2S;
  fbTimer           : TON;
  tT                : T_UtcTime;
END_VAR

fbConnection.Execute();
eState:= fbConnection.eState;

fbTimer(IN:=bSimulation, PT:=tSimulation);
IF fbTimer.Q THEN(* Simulate server data update *)
  fbTimer(IN:=FALSE);
  fbTimer(IN:=bSimulation);

  fbConnection.GetSystemTime(ipAA:=0, tT=>tT);

  (* Simulate "IEDLD1/LLN0.DS1" member value change *)
  fbIED.IEDLD1.LEDGGIO1.SPCS01.t.tValue:=tT;
  fbIED.IEDLD1.LEDGGIO1.SPCS01.stVal.bValue:=NOT fbIED.IEDLD1.LEDGGIO1.SPCS01.stVal.bValue;
  fbIED.IEDLD1.LEDGGIO2.SPCS01.t.tValue:=tT;
  fbIED.IEDLD1.LEDGGIO2.SPCS01.stVal.bValue:=NOT fbIED.IEDLD1.LEDGGIO2.SPCS01.stVal.bValue;
  fbIED.IEDLD1.LEDGGIO3.SPCS01.t.tValue:=tT;
  fbIED.IEDLD1.LEDGGIO3.SPCS01.stVal.bValue:=NOT fbIED.IEDLD1.LEDGGIO3.SPCS01.stVal.bValue;
  fbIED.IEDLD1.LEDGGIO4.SPCS01.t.tValue:=tT;

```



```

fbIED.IEDLD1.LEDGGIO4.SPCS01.stVal.bValue:=NOT fbIED.IEDLD1.LEDGGIO4.SPCS01.stVal.bValue;
fbIED.IEDLD1.LEDGGIO5.SPCS01.t.tValue:=tT;
fbIED.IEDLD1.LEDGGIO5.SPCS01.stVal.bValue:=NOT fbIED.IEDLD1.LEDGGIO5.SPCS01.stVal.bValue;
fbIED.IEDLD1.LEDGGIO6.SPCS01.t.tValue:=tT;
fbIED.IEDLD1.LEDGGIO6.SPCS01.stVal.bValue:=NOT fbIED.IEDLD1.LEDGGIO6.SPCS01.stVal.bValue;
fbIED.IEDLD1.LEDGGIO7.SPCS01.t.tValue:=tT;
fbIED.IEDLD1.LEDGGIO7.SPCS01.stVal.bValue:=NOT fbIED.IEDLD1.LEDGGIO7.SPCS01.stVal.bValue;
fbIED.IEDLD1.LEDGGIO8.SPCS01.t.tValue:=tT;
fbIED.IEDLD1.LEDGGIO8.SPCS01.stVal.bValue:=NOT fbIED.IEDLD1.LEDGGIO8.SPCS01.stVal.bValue;

(* Simulate "IEDLD1/LLN0.DS2" member value change *)
fbIED.IEDLD1.MMXU1.TotW.t.tValue:=tT;
fbIED.IEDLD1.MMXU1.TotW.mag.f.fValue:=fbIED.IEDLD1.MMXU1.TotW.mag.f.fValue + 0.1;

(* Simulate "IEDLD1/LLN0.DS3" member value change *)
fbIED.IEDLD1.XCBR1.Pos.t.tValue:=tT;
IF fbIED.IEDLD1.XCBR1.Pos.stVal.eValue = E_AcsiDbpos.On THEN
    fbIED.IEDLD1.XCBR1.Pos.stVal.eValue:= E_AcsiDbpos.Off;
ELSE
    fbIED.IEDLD1.XCBR1.Pos.stVal.eValue:= E_AcsiDbpos.On;
END IF
fbIED.IEDLD1.XCBR1.Pos.q.OldData:= NOT fbIED.IEDLD1.XCBR1.Pos.q.OldData;
END_IF

(* Execute server control block implementations *)
fbIED.IEDLD1.LLN0.urcb101.Server.Execute();
fbIED.IEDLD1.LLN0.urcb201.Server.Execute();
fbIED.IEDLD1.LLN0.urcb301.Server.Execute();
fbIED.IEDLD1.LLN0.brcb101.Server.Execute();
fbIED.IEDLD1.LLN0.brcb201.Server.Execute();
fbIED.IEDLD1.LLN0.brcb301.Server.Execute();

CASE state OF
  0: (* Initial state *)
    IF _bAbort THEN (* Abort connection => execute AbortReq() command *)
        _bAbort:= FALSE;
        bSuccess:= fbConnection.AbortReq(ipReason:=fbAbortReason, ipSink:=0, ipResult=>ipResult)
    ;
        state:= SEL(bSuccess, 100, 1);
    ELSIF eState = E_AsyncEnvironmentState.Established AND _bDisconnect THEN (* Close/
release connection => execute ReleaseReq() command *)
        _bDisconnect:= FALSE;
        bSuccess:= fbConnection.ReleaseReq(ipSink:=0, ipResult=>ipResult);
        state:= SEL(bSuccess, 100, 1);
    ELSIF eState = E_AsyncEnvironmentState.Established THEN (* Connection established => exchang
e IED data *)
        state:= 10;
    END_IF
    _bDisconnect:= FALSE;

  1: (* Wait for AbortReq() or ReleaseReq() command completion *)
    IF ipResult <> 0 THEN
        ipResult.Execute();
        IF NOT (bBusy:=ipResult.IsBusy()) THEN
            state:= SEL(ipResult.IsCompleted(), 100(* failed or aborted *), 0(* succeeded *));
        END_IF
    END_IF

  10: (* connection established *)
    state:= 0;

  100: (* Error state *)
    state:= 0;
    IF ipResult <> 0 THEN
        sLastErrorResult:= ipResult.Dump();
    END_IF
END_CASE

```

7.25 Server - Ereignisse bei Datenzugriff

Dieses Beispiel zeigt mögliche Anwendungen für die Ereignisse bei Datenzugriff.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1031/TF6510_TC3_IEC61850/Resources/10698267531.zip

Das hier beschriebene Beispiel nutzt die State Machine, welche in dem Kapitel „Allgemeine Server - Projektstruktur [▶ 628]“ beschrieben ist. Die States: 0,1,10 und 100 sind identisch zu der dort beschriebenen State Machine. Andere States wurden für das Beispiel modifiziert oder auch neue States hinzugefügt.

Beim Datenzugriff auf einen Wert im Datenmodell über die Eigenschaft oder Kommunikationsdienste kann durch das Nutzen der Ereignismethoden anwenderspezifisches Verhalten implementiert werden. Dafür kann an jeden Datenobjekt-Funktionsbaustein bestimmte Methoden hinzugefügt werden, die vor und nach Schreib- und Lesezugriffen von der Server-Applikation aufgerufen werden. Dabei handelt es sich um die folgenden vier Methoden:

| Methodenname | Beschreibung |
|----------------|---|
| OnGetVPreEvent | Wird bei Lesezugriffen aufgerufen, bevor die Daten aus dem Datenmodell gelesen werden. In dieser Methode können die zu lesenden Daten angepasst werden oder die Leseanfrage kann mit einem Fehlercode abgelehnt werden. |
| OnGetVEvent | Wird bei Lesezugriffen aufgerufen, nachdem die Daten aus dem Datenmodell gelesen wurden. Diese Methode kann z. B. für das Zurücksetzen von für eine Leseanfrage gesetzten Werten genutzt werden. |
| OnSetVPreEvent | Wird bei Schreibzugriffen aufgerufen, bevor die Daten in das Datenmodell geschrieben werden. In dieser Methode können die zu schreibenden Daten überprüft und die Schreibenanfrage ggfs. abgelehnt werden. |
| OnSetVEvent | Wird bei Schreibzugriffen aufgerufen, nachdem die Daten in das Datenmodell geschrieben wurden. Diese Methode kann z. B. genutzt werden, um andere Werte entsprechend dem geschriebenen Wert zu aktualisieren. |

Im Folgenden werden die in diesem Beispiel implementierten exemplarischen Anwendungsfälle für diese Methoden erläutert.

Substitution

Das Nutzen von Ersatzwerten (Substitution) lässt sich mit den Methoden „OnGetVPreEvent“ und „OnGetVEvent“ umsetzen. Im Beispiel ist dies in den Funktionsbausteinen „FB_DO_IED_LD1_MMXU1_PhV_phsA“ und „FB_DO_IED_LD1_MMXU1_PhV_phsB“ implementiert. Im ersten wird bei aktivierter Substitution bei Leseanfrage der Prozesswert mit dem Ersatzwert überschrieben. Im zweiten Funktionsbaustein wird dabei zudem der aktuelle Prozesswert zwischengespeichert, um diesen nach der Leseanfrage wieder zu setzen.

Prüfen von Einstellungswerten

Im Funktionsbaustein „FB_DO_IED_LD1_CCGR1_OilTmpSet“ ist das Überprüfen von Schreibenanfragen auf Einstellungswerte mit der „OnSetVPreEvent“-Methode implementiert. In dem Funktionsbaustein können über die Datenattribute „minVal“ und „maxVal“ der minimale und maximale gültige Wert für das Datenattribut „setMag“ eingestellt werden. Bei jeder Schreibenanfrage auf „setMag“ wird nun überprüft, ob der zu schreibende Wert innerhalb des eingestellten Bereichs liegt. Ist dies nicht der Fall, wird ein Fehlercode an die Serverapplikation zurückgegeben. Dann wird der Wert nicht geschrieben und der Fehler wird an den Client geschickt.

Runden von Messwerten

In der „OnSetVEvent“-Methode können gerundete Messwerte berechnet und Wertebereich bestimmt werden. Dies ist im Funktionsbaustein „FB_DO_IED_LD1_MMDC1_Vol“ implementiert. Dafür wird im „FB_IEDServerSession“-Funktionsbaustein der Wert des „IEDL1/MMDC1.Vol.instMag.f“-Datenattributs mit einem zufälligen Wert überschrieben. Beim Schreibzugriff über die Eigenschaft wird nach dem Übernehmen des Werts ins Datenmodell die „OnSetVEvent“-Methode aufgerufen. In dieser wird zuerst der Wert vom Datenattribut „instMag.f“ gerundet und in das Datenattribut „mag.f“ übernommen. Danach wird anhand der konfigurierten Wertebereiche bestimmt, in welchem Wertebereich sich der aktuelle Wert von „instMag.f“ befindet. Zudem wird geprüft, ob dieser sich außerhalb des definierten Bereichs befindet.

7.26 Server - Buffered Reporting

Dieses Beispiel zeigt die Buffered-Reporting-Implementierung in einem TwinCAT IEC61850 Serverprojekt. Die für das Reporting benötigten BRCBs (buffered report control blocks) und Datasets können im TwinCAT Telecontrol Configurator konfiguriert oder aus einer vorhandenen SCL-Datei (z. B. ICD-Datei) importiert werden. Die Instanzen der BRCBs und der Datasets werden während der Codegenerierung automatisch angelegt (normalerweise unterhalb von LLN0). Die Initialwerte der BRCB-Attribute können bereits im TwinCAT Telecontrol Configurator konfiguriert werden.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1031/TF6510_TC3_IEC61850/Resources/11740489611.zip

Das hier beschriebene Beispiel nutzt die Statemachine, welche in dem Kapitel „Allgemeine Server - Projektstruktur [▶ 628]“ beschrieben ist. Die States: 0,1,10 und 100 sind identisch zu der dort beschriebenen Statemachine. Andere States wurden für das Beispiel modifiziert oder auch neue States hinzugefügt.

Bei der Implementierung der Buffered-Reporting-Funktionalität in einem TwinCAT IEC 61850 Serverprojekt handelt es sich praktisch um eine „Blackbox“. D.h. die komplette Funktionalität ist in einem Baustein bereits gekapselt und die Applikation muss lediglich diesen Baustein aufrufen. Jeder BRCB besitzt ein solches Baustein-Unterelement mit dem Namen: „Server“. Die SPS-Applikation muss nur die Methode: „Execute“ an dem „Server“-Baustein zyklisch aufrufen. Diese Methode ist für das Versenden der Reports und die Aktualisierung der Statusinformationen in dem dazugehörigen BRCB verantwortlich (siehe Code unten).

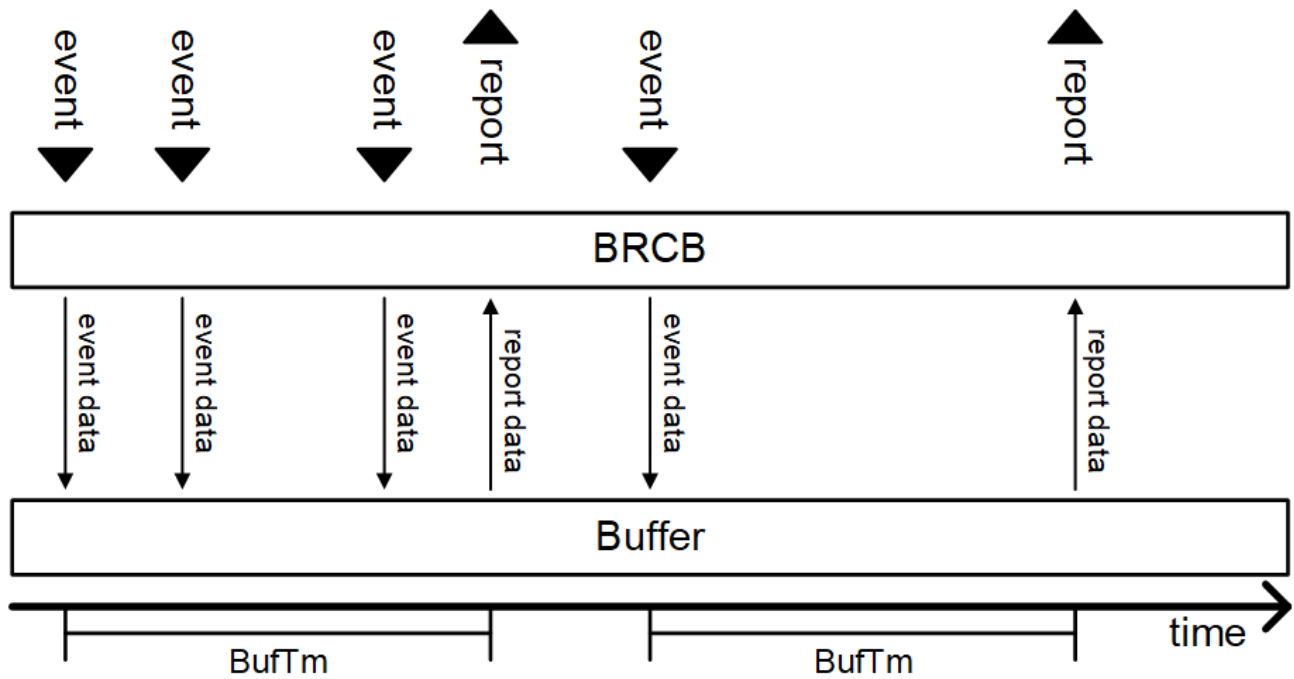
Die Konfiguration der Reports kann entweder auf der Serverseite direkt am BRCB oder seitens Clients via Client-Server-Dienste (GetBRCBDataValues und SetBRCBDataValues) vorgenommen werden. Die via Reporting zu kommunizierenden IED-Daten werden mit Hilfe der Datasets festgelegt. In einem Dataset können mehrere Datenreferenzen (Dataset-Member) konfiguriert werden.

Reports, die nicht versendet werden können, werden im Buffer gespeichert. Ist die maximale Anzahl an Einträgen im Buffer erreicht, wird der älteste Report gelöscht. Bei der Deklaration eines BRCBs kann die maximale Anzahl an Einträgen über die Eigenschaft „nMaxBufferEntries“ eingestellt werden. Im Beispiel ist dies bei der Deklaration von „brcb101“ zu sehen:

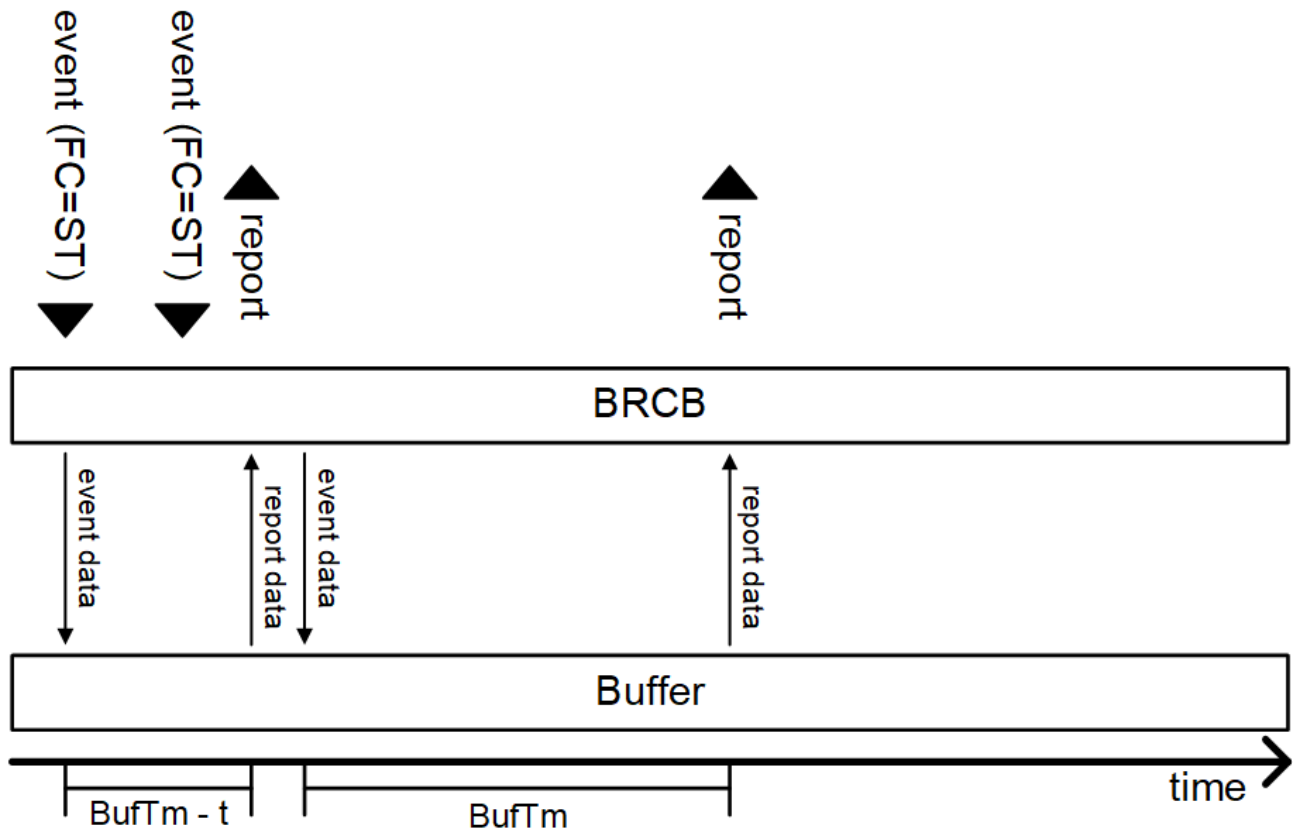
```
brcb101: FB_ScsmBrCBImplClass := (Server:=(nMaxBufferEntries:=15), RptID:=(sValue:='IEDLD1/LLN0.brcb101'), DataSet:=(sValue:='IEDLD1/LLN0.DS1'), ConfRev:=(nValue:=1), OptFlds:=(SequenceNumber:=TRUE, ReportTimeStamp:=TRUE, ReasonForInclusion:=TRUE, DataSetName:=TRUE, DataReference:=TRUE, BufferOverflow:=TRUE, EntryID:=TRUE, ConfRevision:=TRUE), TrgOps:=(DataChange:=TRUE, QualityChange:=TRUE, DataUpdate:=TRUE, Integrity:=TRUE, GeneralInterrogation:=TRUE), IntgPd:=(nValue:=5000), bLinkResult:=THIS^.AddBufferedReportControlBlockToContainer(ipBufferedReportControlBlock:=brcb101));
```

Das Beispiel enthält eine einfache Simulation einiger Dataset-Member-Werte. Wenn die Variable „bSimulation“ auf TRUE gesetzt ist, dann werden die Werte alle 2 Sekunden modifiziert.

Wie erkennt die BRCB-Server-Implementierung, dass ein Wert modifiziert wurde und ein Report generiert werden soll? Dies geschieht durch den Schreibzugriff auf den Attributwert (z. B.: bValue-, tValue-, iValue-, eValue-, nValue-Eigenschaft). Es wird aber nicht sofort bei jedem einzelnen Schreibzugriff auf einen Attributwert ein neuer Report generiert. Dieses wäre unerwünscht. Die „Server“-Implementierung markiert intern die geänderten Dataset-Member als modifiziert. Erst beim Aufruf der „Execute“-Methode wird ein Report generiert und versendet. Die in dem Report versendeten Daten entsprechen den Daten zum Zeitpunkt der Wertänderung des Dataset-Members. Die Werte werden bei Änderung zwischengespeichert und nach Ablauf der im Attribut „BufTm“ konfigurierten Zeit in einen Report kodiert und verschickt. Wenn innerhalb der Bufferzeit ein erneuter Schreibzugriff vorkommt, werden die zwischengespeicherten Daten überschrieben. Dieser Ablauf ist in der folgenden Abbildung dargestellt:



Für Daten der funktionalen Gruppe „ST“ ist in der IEC 61850-7-2 ein abweichendes Verhalten für das Zwischenspeichern definiert. Im Falle, dass vor Ablauf von „BufTm“ der Wert erneut geschrieben wird, werden die zwischengespeicherten Daten sofort zu einem Report kodiert und verschickt. Daraufhin werden die neuen Daten in den Zwischenspeicher geschrieben und die Überwachung der Bufferzeit wird neu gestartet. In der folgenden Grafik ist das beschriebene Verhalten abgebildet:

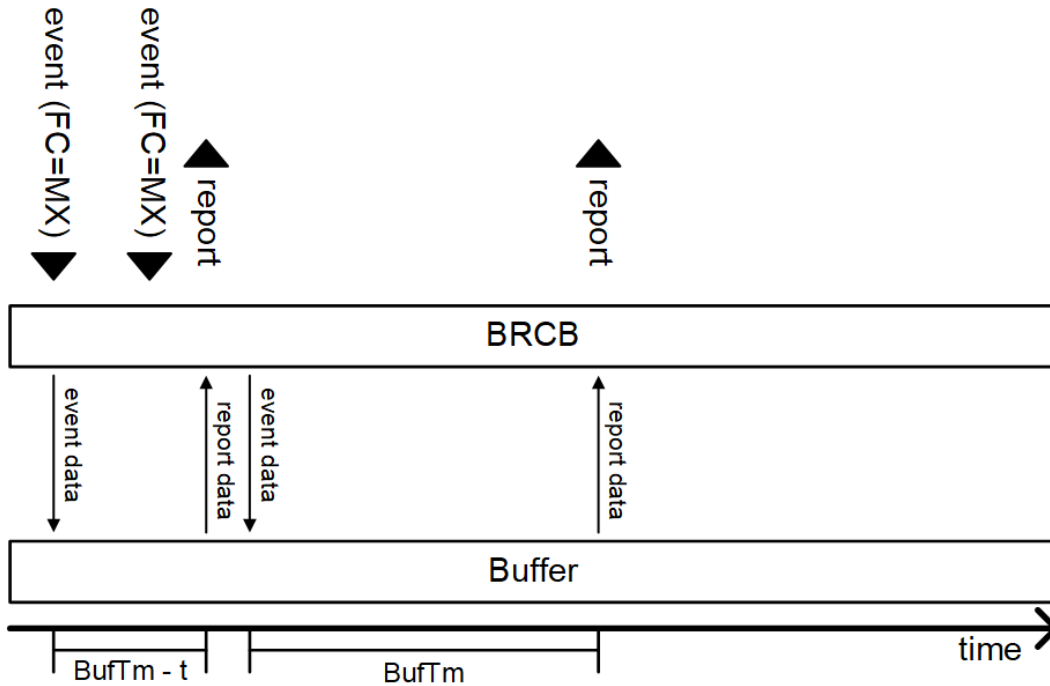


Für Daten der funktionalen Gruppe „MX“ kann ausgewählt werden, ob der BRCB diese gesondert, wie Daten der funktionalen Gruppe „ST“, behandeln soll. Dies kann über die Eigenschaft „bReplaceBufferedMX“ eingestellt werden. Im Beispiel wird die Sonderbehandlung bei „brcb201“ aktiviert:

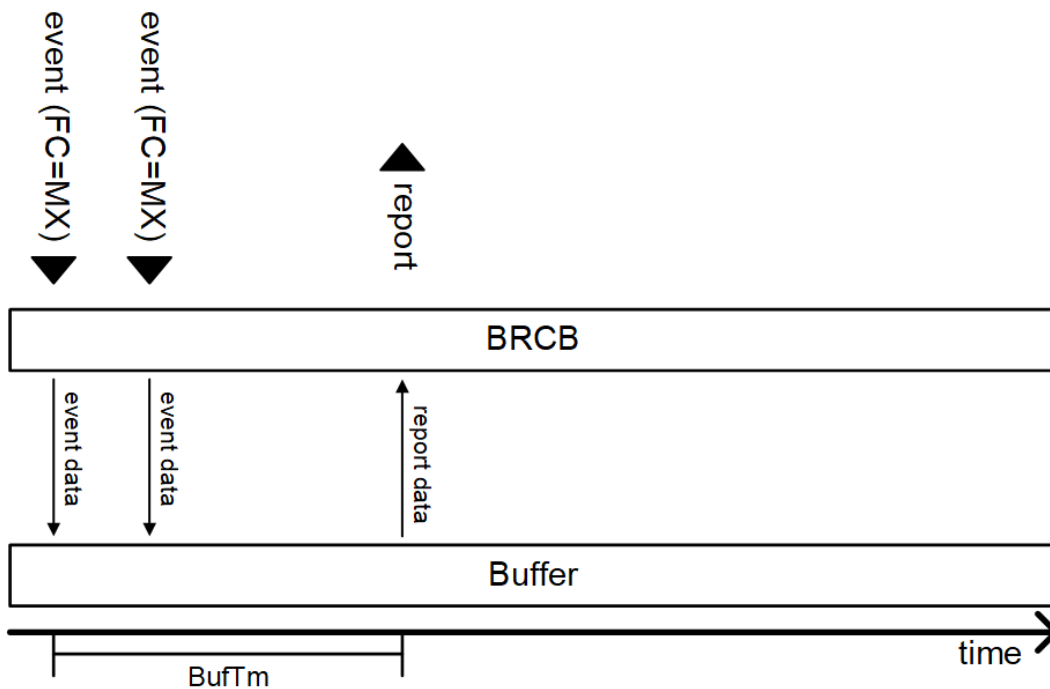
```
brcb201: FB_ScsmBrCBImplClass := (Server:=(bReplaceBufferedMX:=TRUE), RptID:=(sValue:='IEDLD1/LLN0.brCb201'), DataSet:=(sValue:='IEDLD1/LLN0.DS2'), ConfRev:=(nValue:=1), OptFlds:=(SequenceNumber:=TRUE, ReportTimeStamp:=TRUE, ReasonForInclusion:=TRUE, DataSetName:=TRUE, DataReference:=TRUE, BufferOverflow:=TRUE, EntryID:=TRUE, ConfRevision:=TRUE), BufTm:=(nValue:=5000), TrgOps:=(DataChange:=TRUE, QualityChange:=TRUE, DataUpdate:=TRUE, Integrity:=TRUE, GeneralInterrogation:=TRUE), IntgPd:=(nValue:=5000), bLinkResult:=THIS^.AddBufferedReportControlBlockToContainer(ipBufferedReportControlBlock:=brcb201));
```

Wie „MX“ Daten je nach Einstellung von „bReplaceBufferedMX“ behandelt werden ist in der folgen Abbildung zu sehen:

bReplaceBufferedMX = FALSE:



bReplaceBufferedMX = TRUE:



Das sofortige Versenden von Reports bei wiederholter Änderung eines Werts des zugeordneten Datasets kann dazu führen, dass mehr Reports als gewünscht generiert werden. Dies z. B. ist der Fall, wenn ein Datenobjekt der funktionalen Gruppe „ST“ in einem Dataset referenziert ist und dann mehrere Attribute des

Datenobjekts einzeln geschrieben werden. Da jede Wertänderung eines Datenattributs innerhalb eines Datenobjekts auch eine Wertänderung des Datenobjekts darstellt, werden mehrere Reports generiert. Wird die Eigenschaft „bOverwriteSameCycleChanges“ auf TRUE gesetzt, werden mehrfache Wertänderungen innerhalb eines SPS Zyklus als eine einzelne Wertänderung behandelt. Die Erkennung, welche Werte im aktuellen Zyklus geschrieben wurden, wird mit Aufruf der „Execute“-Methode zurückgesetzt. Eine Wertänderung vor und nach dem Aufruf werden daher als nicht als Änderung im selben Zyklus interpretiert.



Beachten Sie, dass beim Schreiben oder Forcen der Werte im TwinCAT Online-Mode keine Reports generiert werden. Die „Server“-Implementierung kann auf diese Weise nicht erkennen, wenn ein Wert modifiziert wurde.

Integrity-Reports und General-Interrogation-Reports werden automatisch entsprechend der Konfiguration des BRCBs generiert und versendet. Das Versenden dieser Reports muss in der SPS-Applikation nicht auf eine besondere Weise getriggert werden. Es wird ebenfalls von dem zyklischen Aufruf der Server-„Execute“-Methode übernommen.

```
PROGRAM MAIN
VAR
  bSimulation          : BOOL:=TRUE; (* Enables/disables simulation of data update/
change *)
  tSimulation          : TIME:=T#2S; (* Cycle time of simulated data update/change *)
  fbTimer              : TON;
  tT                   : T_UtcTime;

  bReplaceBufferedMX  : BOOL; (* Toggles bReplaceBufferedMX property of brcb201 *)
  bOverwriteSameCycleChanges : BOOL; (* Toggles bOverwriteSameCycleChanges property of brcb301
*)
END_VAR

P_IEC61850MAIN();

fbIED.IEDLD1.LLN0.brCb201.Server.bReplaceBufferedMX:= bReplaceBufferedMX;
fbIED.IEDLD1.LLN0.brCb301.Server.bOverwriteSameCycleChanges:= bOverwriteSameCycleChanges;

fbTimer(IN:=bSimulation, PT:=tSimulation);
IF fbTimer.Q THEN(* Simulate server data update *)
  fbTimer(IN:=FALSE);
  fbTimer(IN:=bSimulation);

  fbIEDServer.GetSystemTime(ipAA:=0, tT=>tT);

  (* Simulate "IEDLD1/LLN0.DS1" member value change *)
  fbIED.IEDLD1.LEDGGIO1.SPCS01.t.tValue:= tT;
  fbIED.IEDLD1.LEDGGIO1.SPCS01.stVal.bValue:= NOT fbIED.IEDLD1.LEDGGIO1.SPCS01.stVal.bValue;
  fbIED.IEDLD1.LEDGGIO2.SPCS01.t.tValue:= tT;
  fbIED.IEDLD1.LEDGGIO2.SPCS01.stVal.bValue:= NOT fbIED.IEDLD1.LEDGGIO2.SPCS01.stVal.bValue;
  fbIED.IEDLD1.LEDGGIO3.SPCS01.t.tValue:= tT;
  fbIED.IEDLD1.LEDGGIO3.SPCS01.stVal.bValue:= NOT fbIED.IEDLD1.LEDGGIO3.SPCS01.stVal.bValue;
  fbIED.IEDLD1.LEDGGIO4.SPCS01.t.tValue:= tT;
  fbIED.IEDLD1.LEDGGIO4.SPCS01.stVal.bValue:= NOT fbIED.IEDLD1.LEDGGIO4.SPCS01.stVal.bValue;
  fbIED.IEDLD1.LEDGGIO5.SPCS01.t.tValue:= tT;
  fbIED.IEDLD1.LEDGGIO5.SPCS01.stVal.bValue:= NOT fbIED.IEDLD1.LEDGGIO5.SPCS01.stVal.bValue;
  fbIED.IEDLD1.LEDGGIO6.SPCS01.t.tValue:= tT;
  fbIED.IEDLD1.LEDGGIO6.SPCS01.stVal.bValue:= NOT fbIED.IEDLD1.LEDGGIO6.SPCS01.stVal.bValue;
  fbIED.IEDLD1.LEDGGIO7.SPCS01.t.tValue:= tT;
  fbIED.IEDLD1.LEDGGIO7.SPCS01.stVal.bValue:= NOT fbIED.IEDLD1.LEDGGIO7.SPCS01.stVal.bValue;
  fbIED.IEDLD1.LEDGGIO8.SPCS01.t.tValue:= tT;
  fbIED.IEDLD1.LEDGGIO8.SPCS01.stVal.bValue:= NOT fbIED.IEDLD1.LEDGGIO8.SPCS01.stVal.bValue;

  (* Simulate "IEDLD1/LLN0.DS2" member value change *)
  fbIED.IEDLD1.MMXU1.TotW.t.tValue:= tT;
  fbIED.IEDLD1.MMXU1.TotW.mag.f.fValue:= fbIED.IEDLD1.MMXU1.TotW.mag.f.fValue + 0.1;

  (* Simulate "IEDLD1/LLN0.DS3" member value change *)
  fbIED.IEDLD1.XCBR1.Pos.t.tValue:= tT;
  IF fbIED.IEDLD1.XCBR1.Pos.stVal.eValue = E_AcsiDbpos.On THEN
    fbIED.IEDLD1.XCBR1.Pos.stVal.eValue:= E_AcsiDbpos.Off;
  ELSE
    fbIED.IEDLD1.XCBR1.Pos.stVal.eValue:= E_AcsiDbpos.On;
  END_IF
  fbIED.IEDLD1.XCBR1.Pos.q.OldData:= NOT fbIED.IEDLD1.XCBR1.Pos.q.OldData;
  fbIED.IEDLD1.XCBR1.Pos.q.eValidity:= SEL(fbIED.IEDLD1.XCBR1.Pos.q.OldData, E_AcsiQualityValidity
.Good, E_AcsiQualityValidity.Questionable);
END_IF
```

```

FUNCTION_BLOCK FB_IEDServerSession IMPLEMENTS I_ScsmAbortIndEventSink, I_ScsmAssociateIndEventSink,
I_ScsmReleaseIndEventSink
VAR_INPUT
    fbConnection      : FB_iec61850ConnectionClass := (ipAbortInd:=THIS^, ipAssociateInd:=THIS^, ipRe
leaseInd:=THIS^);
END_VAR
VAR
    _bAbort           : BOOL; (* Rising edge activates AbortReq() command execution *)
    _bDisconnect      : BOOL; (* Rising edge activates ReleaseReq() command execution *)

    state             : BYTE;
    eState             : E_AsyncEnvironmentState; (* Environment state *)
    bBusy              : BOOL; (* If TRUE => command execution is busy *)
    bSuccess           : BOOL;
    ipResult           : I_AsyncServiceResultClass;
    sLastErrorResult  : T_MaxString;
    fbAbortReason     : FB_ServiceErrorClass := (stError:=SUCCESS_EVENT);
    sLastAbortReason  : T_MaxString;
END_VAR

fbConnection.Execute();
eState:= fbConnection.eState;

(* Execute server control block implementations *)
fbIED.IEDLD1.LLN0.brCb101.Server.Execute();
fbIED.IEDLD1.LLN0.brCb201.Server.Execute();
fbIED.IEDLD1.LLN0.brCb301.Server.Execute();

CASE state OF
    0: (* Initial state *)
        IF _bAbort THEN (* Abort connection => execute AbortReq() command *)
            _bAbort:= FALSE;
            bSuccess:= fbConnection.AbortReq(ipReason:=fbAbortReason, ipSink:=0, ipResult=>ipResu
lt);
            state:= SEL(bSuccess, 100, 1);
        ELSIF eState = E_AsyncEnvironmentState.Established AND _bDisconnect THEN (* Close/
release connection => execute ReleaseReq() command *)
            _bDisconnect:= FALSE;
            bSuccess:= fbConnection.ReleaseReq(ipSink:=0, ipResult=>ipResult);
            state:= SEL(bSuccess, 100, 1);
        ELSIF eState = E_AsyncEnvironmentState.Established THEN (* Connection established => excha
nge IED data *)
            state:= 10;
        END_IF
        _bDisconnect:= FALSE;

    1: (* Wait for AbortReq() or ReleaseReq() command completion *)
        IF ipResult <> 0 THEN
            ipResult.Execute();
            IF NOT (bBusy:=ipResult.IsBusy()) THEN
                state:= SEL(ipResult.IsCompleted(), 100(* failed or aborted *), 0(* succeeded *))
            );
        END_IF
    END_IF

    10: (* connection established *)
        state:= 0;

    100: (* Error state *)
        state:= 0;
        IF ipResult <> 0 THEN
            sLastErrorResult:= ipResult.Dump();
        END_IF
END_CASE

```

7.27 Mehrere GOOSE Publisher/Subscriber an einem Netzwerkadapter (ohne Client-Server-Kommunikation)

Dieses Beispiel zeigt die Implementierung von mehreren GOOSE Publishern und Subscribern in einem TwinCAT Projekt. Mehrere Publisher und Subscriber werden zu IED-Gruppen zusammengefasst und so konfiguriert, dass sie für den Austausch der GOOSE-Nachrichten nur einen TwinCAT Real-Time Ethernet Adapter benötigen.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1031/TF6510_TC3_IEC61850/Resources/12863957387.zip

Allgemeine Information zu diesem Beispielprojekt

In einem GOOSE-Netzwerk befinden sich manchmal mehrere baugleiche IED-Geräte (Intelligent Electronic Devices). Diese baugleichen Geräte besitzen dann logischerweise das gleiche IEC 61850 Datenmodell (Logischen-Knoten, Datenobjekte, Datenattribute usw.). Wenn die TwinCAT Applikation mit mehreren baugleichen Geräten GOOSE-Nachrichten austauschen soll, dann ist es von Vorteil im TwinCAT Projekt ein Array von IED-Instanzen für diesen Zweck anzulegen und zu verwenden. Ein IED-Arrayelement entspricht dann einer GOOSE Publisher oder Subscriber-Geräteinstanz im Feld. Die Arrayelemente sollten am besten entweder alle vom Typ: GOOSE Publisher oder alle vom Typ: GOOSE Subscriber sein. Dies vereinfacht die Implementierung der einzelnen GOOSE-Funktionen in der SPS-Applikation. Die SPS-Applikation kann dann z. B. in einer FOR-Schleife alle Publisher aktivieren oder deaktivieren. Andere Konfigurationen (einige IED-Arrayelemente sind Subscriber und andere Publisher) sind auch möglich, werden aber in diesem Beispiel nicht vorgestellt, um das Beispiel einfach halten zu können. Die Geräte im Feld haben logischerweise unterschiedliche IED- und Logical-Device Namen. Alle IED-Arrayelemente haben aber zunächst die gleichen IED- und Logical-Device Namen. Dies stellt für die SPS-Applikation aber kein Problem dar. Die einzelnen IED-Arrayelemente können beim SPS-Programmstart entsprechend umkonfiguriert werden. Jedes IED-Arrayelement bekommt einen neuen, einzigartigen IED- und LD-Namen. Wegen dieser Umbenennung müssen einige Attributwerte, wie z. B. „Gold“ oder „DatSet“ ebenfalls angepasst werden.

Das IEC 61850 Datenmodell für diese baugleichen IED-Geräte ist dann im TwinCAT-Projekt nur einmal vorhanden und muss nicht für jede einzelne IED-Geräteinstanz dupliziert und umbenannt werden. Dies spart Ressourcen und steigert die Performance der Applikation.

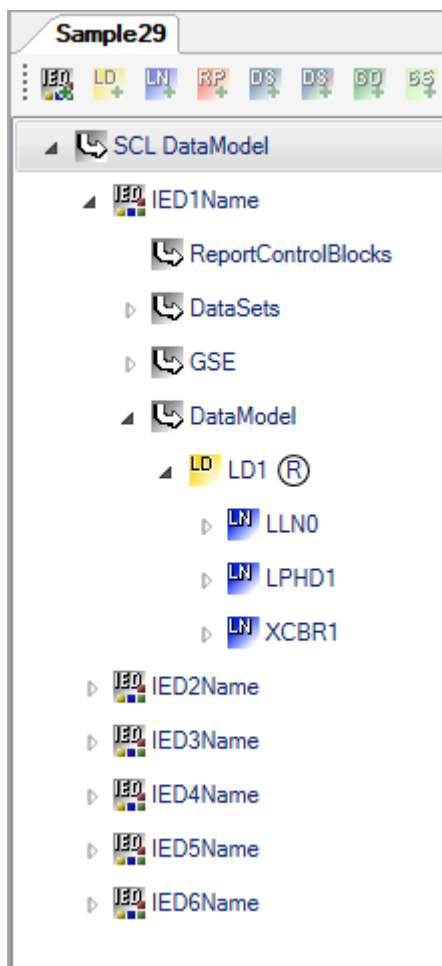
Bei Projekten mit nur einem Subscriber oder Publisher wird der Link von dem „TwinCAT Real-Time Ethernet Adapter“ (RT-Ethernet Adapter) zur SPS über die Instanzen des Funktionsbausteins „FB_[IEDName]Gse“ hergestellt. Das zu kommunizierende IED-Datenmodell eines Subscribers oder Publishers wird dem Funktionsbaustein über die Eigenschaft „ipIED“ mitgeteilt. Weitere Informationen dazu finden Sie unter: [RT – Ethernet Adapter Konfiguration \[► 37\]](#) und in den Beispielen: [Client - GOOSE Subscriber \[► 622\]](#) und [Server - GOOSE Publisher \[► 633\]](#).

Es ist theoretisch möglich bis zu 20 Instanzen des Funktionsbausteins „FB_[IEDName]Gse“ auf diese Weise mit einem „TwinCAT RT-Ethernet Adapter“ zu verlinken. Dadurch können bis zu 20 Publisher oder Subscriber in einem Projekt implementiert werden. Ressourcenschonender ist es aber z. B. nur eine Instanz des Adapterfunktionsbausteins für eine Gruppe von Publishern und eine weitere Instanz für die Gruppe der Subscriber zu verwenden.

Befinden sich alle Geräte in einem gemeinsamen GOOSE-Netzwerk, dann können alle Publisher/Subscriber in der TwinCAT Applikation den gleichen „Real-Time Ethernet Adapter (Multiple Protocol Handler)“ verwenden. Wie ist es im Beispiel realisiert worden? Zuerst wurden mehrere Publisher einer IED-Gruppe zugeordnet. Alle Subscriber wurden einer zweiten IED-Gruppe zugeordnet. Jede IED-Gruppe bekam einen einzigartigen Gruppennamen. Bei den IED-Gruppen handelt es sich um Instanzen des Funktionsbausteins: „FB_GROUP_[GroupName]“. Die zu kommunizierende IED-Gruppe wird dem Adapterfunktionsbaustein „FB_[GroupName]Gse“ über die Eigenschaft „ipGroup“ mitgeteilt. Die zwei Instanzen des Adapterfunktionsbausteins „FB_[GroupName]Gse“ (eine gehört zur Publisher- und die andere zur Subscriber-Gruppe) werden dann mit dem „TwinCAT Real-Time Ethernet Adapter“ wie in der [RT – Ethernet Adapter Konfiguration \[► 37\]](#) beschrieben, verlinkt.

Generierung des IED-Datenmodells

Das IEC 61850 Datenmodell der IEDs in diesem Beispiel wurde zuerst mit dem TwinCAT Telecontrol Configurator aus der SCL-Datei: *Sample29.icd* generiert. Die Datei befindet sich ebenfalls in dem TwinCAT-Projekt zip-Archiv. Diese ICD-Datei beschreibt das Datenmodell von 6 unterschiedlichen IED-Geräten. Im Bild unten sind die IEDs im TwinCAT Telecontrol Configurator zu sehen:



Jedes IED besitzt einen DataSet und einen GOOSE-Steuerblock. Nach der automatischen SPS-Codegenerierung (z.B. als Client-Projekt) wurde das SPS-Projekt manuell modifiziert und die nicht benötigten Funktionsbausteine für die Client/Server-Kommunikation entfernt.

Definition der IED-Gruppen

Aus den 6 automatisch generierten IED-Datenmodellen sollen in diesem Beispielprojekt insgesamt 3 Publisher und 11 Subscriber implementiert werden. Alle Publisher wurden einer IED-Gruppe mit dem Namen „PubGroupName“ und die Subscriber einer IED-Gruppe mit dem Namen „SubGroupName“ zugeordnet.

Globale Variablenliste TcTelecontrol

In dieser GVL wird jeweils eine Publisher-Gruppe „fbPubGroupName“ und eine Subscriber-Gruppe „fbSubGroupName“ instanziiert. Außerdem befindet sich dort eine Instanz des Publisher-Adapterfunktionsbausteins „fbPubGroupNameGse“ und eine Instanz des Subscriber-Adapterfunktionsbausteins „fbSubGroupNameGse“.

Mit Hilfe der Eigenschaft „ipGroup“ wird dem „fbPubGroupNameGse“ die Publisher IED-Gruppe zugewiesen und dem „fbSubGroupNameGse“ die Subscriber IED-Gruppe zugewiesen.

```
VAR_GLOBAL
  ipCreator: I_AcsiCodeCreatorClass := GVL_AcsiVars.Creator.SetCodeRev(codeRev:=3).SetGuiVer(major
:=3, minor:=1, build:=96, revision:=3);

  fbSubGroupName: FB_GROUP_SubGroupName:=(bLinkResult:=fbSubGroupName.Init());
  fbPubGroupName: FB_GROUP_PubGroupName:=(bLinkResult:=fbPubGroupName.Init());

  fbSubGroupNameGse: FB_SubGroupNameGse := (fbAdapter:=(ipGroup:=fbSubGroupName, settings:=(sMulti
castAddr:='01-0C-CD-01-00-01', eDispatchMode:=E_GseDispatchMode.NonPromiscuous)));
  fbPubGroupNameGse: FB_PubGroupNameGse := (fbAdapter:=(ipGroup:=fbPubGroupName, settings:=(sMulti
castAddr:='01-0C-CD-01-00-00', eDispatchMode:=E_GseDispatchMode.NonPromiscuous)));
END_VAR
```


Subscriber Gruppe

Die Subscriber-Gruppe „FB_GROUP_SubGroupName“ konfiguriert 11 Subscriber. „IED5Name“ wird als Array[1..8] in der Subscriber Gruppe instanziiert. Jedes Arrayelement soll ein Subscriber sein, der GOOSE-Nachrichten von einem anderen baugleichen Gerät im Feld empfängt. Die Subscriber-Gruppe kann nur GOOSE-Nachrichten von einem Publisher empfangen, wenn dieser die GOOSE-Nachricht an die Adresse: '01-0C-CD-01-00-01' oder Broadcastadresse oder an die MAC-Adresse des TwinCAT RT-Netzwerkadapters sendet. Die Subscriber-Quelladresse wird für alle Subscriber in einer Subscriber-Gruppe an der Instanz des „FB_[GroupName]Gse“ konfiguriert. Die Tabelle unten listet die wichtigsten Parameter der Subscriber in dieser IED-Gruppe auf.

| IED | IED-Name/LD-Name | SrcAddress | APPID | GoID | DatSet |
|-------------|----------------------|-------------------|---------|---------------------------|-------------------------|
| IED1Name | IED1Name/IED1NameLD1 | 01-0C-CD-01-00-01 | 16#0010 | 'IED1NameLD1/LLN0.gocb01' | 'IED1NameLD1/LLN0.ds01' |
| IED2Name | IED2Name/IED2NameLD1 | 01-0C-CD-01-00-01 | 16#0020 | 'IED2NameLD1/LLN0.gocb01' | 'IED2NameLD1/LLN0.ds01' |
| IED3Name | IED3Name/IED3NameLD1 | 01-0C-CD-01-00-01 | 16#0030 | 'IED3NameLD1/LLN0.gocb01' | 'IED3NameLD1/LLN0.ds01' |
| IED5Name[1] | CH1/CH1LD1 | 01-0C-CD-01-00-01 | 16#0051 | 'CH1LD1/LLN0.GoID.gocb01' | 'CH1LD1/LLN0.ds01' |
| IED5Name[2] | CH2/CH2LD1 | 01-0C-CD-01-00-01 | 16#0052 | 'CH2LD1/LLN0.GoID.gocb01' | 'CH2LD1/LLN0.ds01' |
| IED5Name[3] | CH3/CH3LD1 | 01-0C-CD-01-00-01 | 16#0053 | 'CH3LD1/LLN0.GoID.gocb01' | 'CH3LD1/LLN0.ds01' |
| IED5Name[4] | CH4/CH4LD1 | 01-0C-CD-01-00-01 | 16#0054 | 'CH4LD1/LLN0.GoID.gocb01' | 'CH4LD1/LLN0.ds01' |
| IED5Name[5] | CH5/CH5LD1 | 01-0C-CD-01-00-01 | 16#0055 | 'CH5LD1/LLN0.GoID.gocb01' | 'CH5LD1/LLN0.ds01' |
| IED5Name[6] | CH6/CH6LD1 | 01-0C-CD-01-00-01 | 16#0056 | 'CH6LD1/LLN0.GoID.gocb01' | 'CH6LD1/LLN0.ds01' |
| IED5Name[7] | CH7/CH7LD1 | 01-0C-CD-01-00-01 | 16#0057 | 'CH7LD1/LLN0.GoID.gocb01' | 'CH7LD1/LLN0.ds01' |
| IED5Name[8] | CH8/CH8LD1 | 01-0C-CD-01-00-01 | 16#0058 | 'CH8LD1/LLN0.GoID.gocb01' | 'CH8LD1/LLN0.ds01' |

Der Group-Funktionsbaustein „FB_GROUP_SubGroupName“ ist abgeleitet von dem Funktionsbaustein „FB_AcsiCommonIEDGroupClass“ und dient als Container für mehrere IEDs. Die Methode „AddIEDToContainer“ fügt die einzelnen IEDs der Subscriber-Gruppe hinzu. Bei den IED-Arrayelementen muss auf diese Weise jedes einzelne Arrayelement der Subscriber-Gruppe hinzugefügt werden.

Die Eigenschaft „Tag“ kann verwendet werden, um die einzelnen Arrayelemente von „IED5Name“ mit einer zusätzlichen Information zu versehen. Im Beispiel sind dort die Array-Elementnummern gesetzt. Diese Nummer kann in der „Init“-Methode oder später zur Laufzeit in den Methoden des IEDs abgefragt werden. Jedes Arrayelement kann auch abhängig von der Nummer, ein wenig anders konfiguriert werden.

```
FUNCTION_BLOCK FB_GROUP_SubGroupName EXTENDS FB_AcsiCommonIEDGroupClass
VAR CONSTANT
    Max_GROUP_IED5Name: DINT:=8; (* Max. number of IED5Name instances *)
END_VAR
VAR_INPUT
    IED1Name: FB_IED_IED1Name:=(bLinkResult:=THIS^.AddIEDToContainer(ipIED:=IED1Name));
    IED2Name: FB_IED_IED2Name:=(bLinkResult:=THIS^.AddIEDToContainer(ipIED:=IED2Name));
    IED3Name: FB_IED_IED3Name:=(bLinkResult:=THIS^.AddIEDToContainer(ipIED:=IED3Name));
    IED5Name: ARRAY[1..Max_GROUP_IED5Name] OF FB_IED_IED5Name:=
    [
        (Tag:=1, sObjectName:='CH1', bLinkResult:=THIS^.AddIEDToContainer(ipIED:=IED5Name[1])),
        (Tag:=2, sObjectName:='CH2', bLinkResult:=THIS^.AddIEDToContainer(ipIED:=IED5Name[2])),
        (Tag:=3, sObjectName:='CH3', bLinkResult:=THIS^.AddIEDToContainer(ipIED:=IED5Name[3])),
        (Tag:=4, sObjectName:='CH4', bLinkResult:=THIS^.AddIEDToContainer(ipIED:=IED5Name[4])),
        (Tag:=5, sObjectName:='CH5', bLinkResult:=THIS^.AddIEDToContainer(ipIED:=IED5Name[5])),
        (Tag:=6, sObjectName:='CH6', bLinkResult:=THIS^.AddIEDToContainer(ipIED:=IED5Name[6])),
        (Tag:=7, sObjectName:='CH7', bLinkResult:=THIS^.AddIEDToContainer(ipIED:=IED5Name[7])),
        (Tag:=8, sObjectName:='CH8', bLinkResult:=THIS^.AddIEDToContainer(ipIED:=IED5Name[8]))
    ];

```

```

END_VAR
VAR
    fbError      : FB_ServiceErrorClass;
    ipLastError  : I_ServiceErrorClass;
    i            : DINT; (* Temp array index variable *)
END_VAR

```

Methoden der Subscriber-Gruppe

Der Funktionsbaustein „FB_GROUP_SubGroupName“ implementiert mehrere Methoden. „Enable“ aktiviert alle Subscriber. „Disable“ deaktiviert alle Subscriber. „Execute“ muss zyklisch aufgerufen werden. „Init“ initialisiert alle IEDs, LDs, LNs usw. in der Gruppe. Die Methode „Init“ wird vom Initialisierungscode (bereits vor dem ersten SPS-Zyklus) einmalig ausgeführt.

Anpassung des IED-Namens und LD-Namens beim IED5Name Array

Der Setter der Eigenschaft „sObjectName“ des Funktionsbausteins „FB_IED_IED5Name“ wurde überschrieben. Beim Setzen eines neuen IED-Namens wird die neue Setter-Implementierung aufgerufen und der neue IED-Name übernommen. Der Logical-Device-Name des Funktionsbausteins „FB_LD_IED5NameLD1“ wird ebenfalls korrigiert da dieser auch den IED-Namen enthält.

```

PROPERTY FINAL sObjectName : T_AcsiObjectName
VAR
END_VAR

(* Adopt the modified IED name *)
SUPER^.sObjectName:= sObjectName;

(* Adjust the logical-device names too *)
IED6NameLD1.sObjectName:= CONCAT(sObjectName, 'LD1');

```

Anpassung der APPID, GoID und DataSet beim IED5Name Array

Die Anpassung wird in der Methode „Init“ des logischen Knotens „LLN0“ vorgenommen.

```

METHOD FINAL Init : BOOL
VAR_INPUT
    IED: REFERENCE TO FB_IED_IED5Name;
    LogicalDevice: REFERENCE TO FB_LD_IED5NameLD1;
END_VAR
VAR
END_VAR

IF NOT __ISVALIDREF(IED) THEN RETURN; END_IF
IF NOT __ISVALIDREF(LogicalDevice) THEN RETURN; END_IF

(* Example: Adjust subscribers GoID string *)
gocb01.GoID.sValue:= CONCAT(CONCAT(LogicalDevice.sObjectName, '/'), CONCAT(CONCAT(THIS^.sObjectName, '.'), gocb01.sObjectName));

(* Example: Adjust subscribers data-set reference string *)
gocb01.DatSet.sValue:= CONCAT(CONCAT(LogicalDevice.sObjectName, '/'), CONCAT(CONCAT(THIS^.sObjectName, '.'), ds01.sObjectName));

(* Example: Adjust subscribers APPID by use of IED's Tag property value *)
gocb01.DstAddress.APPID.nValue:= WORD#16#0050 + TO_WORD(IED.Tag);

(* Forward IED reference to all data-sets *)
Init:= TRUE;
Init:= Init AND ds01.Init(IED:=IED);

```

Publisher Gruppe

Die Publisher-Gruppe „FB_GROUP_PubGroupName“ konfiguriert 3 Publisher. „IED6Name“ wird als Array[1..2] in der Publisher Gruppe instanziiert. Jedes Arrayelement soll ein Publisher sein, der die GOOSE-Nachrichten an eine andere Zieladresse und mit einer speziellen APPID sendet. Die Tabelle unten listet die wichtigsten Parameter der Publisher in dieser Gruppe auf.

| IED | IED-Name/LD-Name | DstAddress | APPID | GoID | DatSet |
|-------------|--------------------------|-----------------------|---------|-------------------------------|-----------------------------|
| IED4Name | IED4Name/
IED4NameLD1 | 01-0C-
CD-01-00-00 | 16#0040 | 'IED4NameLD1/
LLN0.gocb01' | 'IED4NameLD1/
LLN0.ds01' |
| IED6Name[1] | CNT1/CNT1LD1 | 01-0C-
CD-01-00-01 | 16#0061 | 'CNT1LD1/
LLN0.gocb01' | 'CNT1LD1/LLN0.ds01' |
| IED6Name[2] | CNT2/CNT2LD1 | 01-0C-
CD-01-00-02 | 16#0062 | 'CNT2LD1/
LLN0.gocb01' | 'CNT2LD1/LLN0.ds01' |

Der Group-Funktionsbaustein „FB_GROUP_PubGroupName“ ist abgeleitet von dem Funktionsbaustein „FB_AcsiCommonIEDGroupClass“ und dient als Container für mehrere IEDs. Die Methode „AddIEDToContainer“ fügt die einzelnen IEDs der Publisher-Gruppe hinzu. Bei den IED-Arrayelementen muss auf diese Weise jedes einzelne Arrayelement der Publisher-Gruppe hinzugefügt werden.

Die Eigenschaft „Tag“ kann verwendet werden, um die einzelnen Arrayelemente von „IED6Name“ mit einer zusätzlichen Information zu versehen. Im Beispiel sind dort die Array-Elementnummern gesetzt. Diese Nummer kann in der „Init“-Methode oder später zur Laufzeit in den Methoden des IEDs abgefragt werden. Jedes Arrayelement kann auch, abhängig von der Nummer, ein wenig anders konfiguriert werden.

```

FUNCTION_BLOCK FB_GROUP_PubGroupName EXTENDS FB_AcsiCommonIEDGroupClass
VAR CONSTANT
    Max_GROUP_IED6Name: DINT:=2; (* Max. number of IED6Name instances *)
END_VAR
VAR_INPUT
    IED4Name: FB_IED_IED4Name:=(bLinkResult:=THIS^.AddIEDToContainer(ipIED:=IED4Name));
    IED6Name: ARRAY[1..Max_GROUP_IED6Name] OF FB_IED_IED6Name:=
    [
        (Tag:=1, sObjectName:='CNT1', bLinkResult:=THIS^.AddIEDToContainer(ipIED:=IED6Name[1])),
        (Tag:=2, sObjectName:='CNT2', bLinkResult:=THIS^.AddIEDToContainer(ipIED:=IED6Name[2]))
    ];
END_VAR
VAR
    bUpdate_GROUP_IED4Name_IED4NameLD1_LLN0_gocb01: BOOL; (* If TRUE => Update 'IED4Name.IED4NameLD1
    .LLN0.gocb01' publisher *)
    aUpdate_GROUP_IED6Name_IED6NameLD1_LLN0_gocb01: ARRAY[1..Max_GROUP_IED6Name] OF BOOL; (* If arra
    y element n is TRUE => Update 'IED6Name[n].IED6NameLD1.LLN0.gocb01' publisher *)

    fbError          : FB_ServiceErrorClass;
    ipLastError      : I_ServiceErrorClass;
    i                 : DINT; (* Temp array index variable *)
    iUpdateElement   : DINT; (* Array index of first array element that needs to be updated *)

    bSimulation      : BOOL := TRUE; (* Enables/disables simulation of cyclic GOOSE data update *)
    tSimulation       : TIME := T#5S; (* Cycle time of simulated GOOSE data update *)
    fbUpdateTimer     : TON;
    iLiveCounter      : LINT; (*Live counter (example) *)
    tT                : T_UtcTime; (* Temp time-stamp value *)
END_VAR
    
```

Methoden der Publisher-Gruppe

Der Funktionsbaustein „FB_GROUP_PubGroupName“ implementiert mehrere Methoden. „Start“ startet alle Publisher. „Stop“ stoppt alle Publisher. „Execute“ muss zyklisch aufgerufen werden. „Update“ führt ein Update der Publisher-Daten. „Init“ Initialisiert alle IEDs, LDs, LNs usw. in der Gruppe. Die Methode „Init“ wird vom Initialisierungscode (bereits vor dem ersten SPS-Zyklus) einmalig ausgeführt. In der Methode „MapSimuPubData“ werden Publisher-Daten simuliert/verändert.

Anpassung des IED-Namens und LD-Namens beim IED6Name Array

Der Setter der Eigenschaft „sObjectName“ des Funktionsbausteins „FB_IED_IED6Name“ wurde überschrieben. Beim Setzen eines neuen IED-Namens wird die neue Setter-Implementierung aufgerufen und der neue IED-Name übernommen. Der Logical-Device-Name des Funktionsbausteins „FB_LD_IED6NameLD1“ wird ebenfalls korrigiert da dieser auch den IED-Namen enthält.

```

PROPERTY FINAL sObjectName : T_AcsiObjectName
VAR
END_VAR

(* Adopt the modified IED name *)
SUPER^.sObjectName:= sObjectName;

(* Adjust the logical-device names too *)
IED6NameLD1.sObjectName:= CONCAT(sObjectName, 'LD1');
    
```

Anpassung der APPID, GoID und DataSet beim IED6Name Array

Die Anpassung findet statt in der Methode „Init“ des logischen Knotens „LLN0“.

```
METHOD FINAL Init : BOOL
VAR_INPUT
    IED: REFERENCE TO FB_IED_IED6Name;
    LogicalDevice: REFERENCE TO FB_LD_IED6NameLD1;
END_VAR
VAR
    mac: ETHERNET_ADDRESS;
END_VAR

IF NOT __ISVALIDREF(IED) THEN RETURN; END_IF
IF NOT __ISVALIDREF(LogicalDevice) THEN RETURN; END_IF

(* Example: Adjust publishers GoID string *)
gocb01.GoID.sValue:= CONCAT(CONCAT(LogicalDevice.sObjectName, '/'), CONCAT(CONCAT(THIS^.sObjectName,
    '.'), gocb01.sObjectName));

(* Example: Adjust publishers data-set reference string *)
gocb01.DatSet.sValue:= CONCAT(CONCAT(LogicalDevice.sObjectName, '/'), CONCAT(CONCAT(THIS^.sObjectName,
    e, '.'), ds01.sObjectName));

(* Example: Adjust publishers multicast address by use of IED's Tag property value *)
mac:= THIS^.gocb01.DstAddress.macAddr;// Get current address
mac.b[5]:= TO_BYTE(IED.Tag);// Modify last digit
THIS^.gocb01.DstAddress.macAddr:= mac;// Set new address

(* Example: Adjust publishers APPID by use of IED's Tag property value *)
gocb01.DstAddress.APPID.nValue:= WORD#16#0060 + TO_WORD(IED.Tag);

(* Forward IED reference to all data-sets *)
Init:= TRUE;
Init:= Init AND ds01.Init(IED:=IED);
```

Im Beispielcode oben wird die letzte Stelle der MAC-Adresse von „DstAddress“ gleich dem Wert der „Tag“-Eigenschaft gesetzt. Falls eine komplett andere MAC-Adresse für die zwei Publisher konfiguriert werden soll dann kann die „Tag“-Eigenschaft auch auf folgende Weise verwendet werden:

```
IF IED.Tag = 1 THEN
    THIS^.gocb01.DstAddress.sAddr:='01-0C-CD-01-02-56';
ELSIF IED.Tag = 2 THEN
    THIS^.gocb01.DstAddress.sAddr:='01-0C-CD-01-01-78';
END_IF
```

Netzwerkadapter-Funktionsbaustein FB_SubGroupNameGse

Eine steigende Flanke an der booleschen Variablen „bSubscribe“ (Initialwert ist bereits „TRUE“) aktiviert beim Start der SPS alle Subscriber in der Subscriber-Gruppe. Eine steigende Flanke an der booleschen Variablen „bUnsubscribe“ deaktiviert alle Subscriber in der Gruppe. In der übrigen Zeit muss die „Execute“ Methode der Subscriber-Gruppe und des Netzwerkadapters zyklisch aufgerufen werden. Dies ist notwendig damit die GOOSE-Nachrichten empfangen und an die Subscriber-Instanzen weitergegeben werden.

```
FUNCTION_BLOCK FB_SubGroupNameGse IMPLEMENTS I_GseLinkStatusEventSink
VAR_INPUT
    fbAdapter : FB_GseAdapterClass := (ipLinkStatus:=THIS^);
END_VAR
VAR
    eLinkStatus : E_GseLinkStatus;
    bSuccess : BOOL;
    ipError : I_ServiceErrorClass;
    bSubscribe : BOOL := TRUE; (* Rising edge enables subscriber GoCB's *)
    bUnsubscribe: BOOL; (* Rising edge disables subscriber GoCB's *)
END_VAR

bSuccess:= fbAdapter.Execute(ipError=>ipError); (* Execute network adapter *)

IF bSubscribe THEN (* Enable subscriber GoCB's *)
    bSubscribe:= FALSE;
    bSuccess:= fbSubGroupName.Enable(ipAdapter:=fbAdapter, ipError=>ipError);
ELSIF bUnsubscribe THEN (* Disable subscriber GoCB's *)
    bUnsubscribe:= FALSE;
    bSuccess:= fbSubGroupName.Disable(ipError=>ipError);
ELSE (* Execute subscriber GoCB's *)
    bSuccess:= fbSubGroupName.Execute(ipError=>ipError);
END_IF
```

Netzwerkadapter-Funktionsbaustein FB_PubGroupNameGse

Eine steigende Flanke an der booleschen Variablen „bStart“ (Initialwert ist bereits „TRUE“) aktiviert beim Start der SPS alle Publisher in der Publisher-Gruppe. Eine steigende Flanke an der booleschen Variablen „bStop“ deaktiviert alle Publisher in der Gruppe. Bei einer steigenden Flanke an der booleschen Variablen „bUpdate“ wird die „Update“-Methode der Publisher-Gruppe aufgerufen. Die „bUpdate“-Variable wird aber nur „TRUE“ wenn die „MapSimuPubData“-Methode auch „TRUE“ zurückliefert, weil die Daten von mindestens einen Publisher geändert oder aktualisiert wurden. In der übrigen Zeit muss die „Execute“-Methode der Publisher-Gruppe und des Netzwerkadapters zyklisch aufgerufen werden. Dies ist notwendig damit die Publisher-Instanzen ihre GOOSE-Nachrichten senden können.

```

FUNCTION_BLOCK FB_PubGroupNameGse IMPLEMENTS I_GseSystemClockEventSink, I_GseLinkStatusEventSink
VAR_INPUT
    fbAdapter    : FB_GseAdapterClass := (ipSystemClock:=THIS^, ipLinkStatus:=THIS^);
END_VAR
VAR
    eLinkStatus : E_GseLinkStatus;
    bSuccess    : BOOL;
    ipError     : I_ServiceErrorClass;
    bStart      : BOOL := TRUE; (* Rising edge starts publishing of GOOSE data *)
    bStop       : BOOL; (* Rising edge stops publishing of GOOSE data *)
    bUpdate     : BOOL; (* Rising edge updates published GOOSE data *)
    bSync       : BOOL := TRUE; (* Rising edge synchronizes the GSE system clock to external time *)
    tSync       : T_UtcTime := String_TO_UtcTime(in:='UT#2019-07-12-12:00:00.000000000|000|
3'); (* External (test) time *)
    fbClock     : FB_GseSystemClock;
END_VAR

IF bSync THEN (* Synchronize clock time to external time source *)
    bSync:= FALSE;
    bSuccess:= fbClock.SetToUtcTime(in:=tSync);
ELSE (* Execute clock instance *)
    fbClock.Execute();
END_IF

(* Map/simulate GOCB's data-set member values: 'IED4Name.IED4NameLD1.LLN0.ds01' and/
or 'ARRAY OF IED6Name[1..Max_GROUP_IED6Name].IED6NameLD1.LLN0.ds01' *)
bUpdate:= fbPubGroupName.MapSimuPubData(ipClock:=fbClock);

bSuccess:= fbAdapter.Execute(ipError=>ipError); (* Execute network adapter *)

IF bStart THEN (* Start publishing of GOOSE data *)
    bStart:= FALSE;
    bSuccess:= fbPubGroupName.Start(ipAdapter:=fbAdapter, ipError=>ipError);
ELSIF bStop THEN (* Stop publishing of GOOSE data *)
    bStop:= FALSE;
    bSuccess:= fbPubGroupName.Stop(ipError=>ipError);
ELSIF bUpdate THEN (* Update published GOOSE data *)
    bUpdate:= FALSE;
    bSuccess:= fbPubGroupName.Update(ipError=>ipError);
ELSE (* Execute publisher GoCB's *)
    bSuccess:= fbPubGroupName.Execute(ipError=>ipError);
END_IF

```

Simulation der Publisher-Daten im Beispielprojekt

Wenn die boolesche Variable „bSimulation“ den Wert „TRUE“ hat, dann werden in der Methode „MapSimuPubData“ einige sich verändernde Publisher-Daten simuliert. Z. B. ein Zählerwert „iLiveCounter“ oder einige Quality-Flags.

Zusätzlich wurde dort zu Testzwecken ein sehr einfacher Datenkonzentrator implementiert. Einige Subscriber-Daten der IEDs der Subscriber-Gruppe werden bei Wertänderung in die Publisher-Daten umkopiert. Bei einer Wertänderung wird zusätzlich noch eine boolesche Variable wie z. B. „bUpdate_GROUP_IED4Name_IED4NameLD1_LLN0_gocb01“ auf „TRUE“ gesetzt. Die „Update“-Methode kann über diese Variable erkennen, dass ein Update der Publisher-Daten des Steuerblocks „gocb01“ durchgeführt werden soll.

Test der Subscriber-Gruppe

In dem Zip-Archiv befinden sich noch weitere ICD-Dateien, die zu Test- oder Simulationszwecken der Subscriber-Gruppe verwendet werden können:

Publisher_IED1Name.icd
Publisher_IED2Name.icd
Publisher_IED3Name.icd
Publisher_CH1_To_CH8.icd

Diese Dateien können Sie in einem Dritthersteller Tool zur Simulation der „IED1Name“, „IED2Name“, „IED3Name“ und „CH1“..„CH8“ Publisher verwenden. Außerdem können Sie aus diesen Dateien auch ein TwinCAT Publisher-Projekt erstellen. Die Subscriber in diesem Beispielprojekt können die Daten dieser Publisher empfangen.

Test der Publisher-Gruppe

In dem Zip-Archiv befinden sich noch zwei weitere ICD-Dateien, die beim Test der Publisher-Gruppe in diesem Beispielprojekt verwendet werden können:

Subscriber_CNT1_To_CNT2.icd
Subscriber_IED4Name.icd

Diese Dateien können Sie in einem Dritthersteller Tool zur Simulation der „CNT1“.. „CNT8“ und „IED4Name“ Subscriber verwenden. Die gesendeten Daten der Publisher-Gruppe in diesem Beispielprojekt können von diesen Subscribern empfangen werden.

7.28 TLS – Sichere Client-Server-Kommunikation mit Hilfe von Zertifikaten oder PSK (Pre-Shared Key)

Dieses Beispiel zeigt, wie eine sichere Datenkommunikation mit Hilfe von TLS (Transport Layer Security) in einer TwinCAT IEC 61850 Client- und Server-Applikation realisiert werden kann. Das TLS-Protokoll erlaubt nur die Verschlüsselung der Client-Server-Kommunikation. Die GOOSE-Publisher/Subscriber-Daten können auf diese Weise nicht verschlüsselt werden. Wenn sie eine ältere Version des TwinCAT IEC 61850 Telecontrol Configurators verwenden, dann wird dort die TLS-Funktionalität möglicherweise noch nicht unterstützt. Um dieses Beispielprojekt fehlerfrei übersetzen und testen zu können benötigen Sie eine neuere Version des TF6510 IEC 61850 Telecontrol.

Systemvoraussetzung

TF6510 IEC 61850 Telecontrol v3.1.96.7 oder neuer

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1031/TF6510_TC3_IEC61850/Resources/13651402763.zip

Allgemeine Information zu diesem Beispielprojekt

Das TLS-Beispiel besteht aus zwei separaten TwinCAT IEC 61850 Projekten. Nach dem Auspacken des Zip-Archives finden Sie im Unterordner `\ClientWithTLS\Sample30` ein TwinCAT IEC 61850 Client-Projekt und im Unterordner `\ServerWithTLS\Sample30` ein TwinCAT IEC 61850 Server-Projekt.

Das hier beschriebene: [Client – Basis Beispielprojekt \[► 596\]](#) und hier beschriebene: [Server – Basis Beispielprojekt \[► 632\]](#) dienen als Grundlage für das TLS-Beispielprojekt.

Dieses Beispiel zeigt hauptsächlich zwei Möglichkeiten der TLS-Datenverschlüsselung:

1. TLS mit Zertifikaten (z. B.: eigene, selbstsignierte Zertifikate).
2. TLS mit PSK (Pre-Shared Key).

Jedes TwinCAT IEC 61850 Client-/Server-Projekt kann auf relativ einfache Weise um die TLS-Funktionalität erweitert werden. Damit die Verschlüsselung/Entschlüsselung der Daten auf beiden Seiten funktioniert, muss dies auch logischerweise auf beiden Seiten, beim Client und beim Server erfolgen. Diese Erweiterung besteht hauptsächlich aus der Konfiguration der privaten Schlüssel (Private-Keys), Passwörter, Zertifikate oder der PSK-Schlüssel. Bei der Konfiguration der privaten Schlüssel und Zertifikate werden z.B. dem Server- bzw. Client-Funktionsbaustein die Dateipfade zu den Ordnern mitgeteilt, in denen die privaten Schlüsseldateien und Zertifikate abgelegt wurden. Die benötigten TLS-Konfigurationsparameter werden dabei vorher in einer dafür vorgesehenen Funktionsbausteininstanz vom Typ: `FB_SocketTlsSettingsClass` [\[► 204\]](#) gespeichert. Die Übergabe dieser Information an die IEC 61850 Client/Server-Funktionsbausteine

erfolgt über eine definierte Schnittstelle vom Typ `I_SocketTlsSettingsClass` [► 331], welche der `FB_SocketTlsSettingsClass` [► 204]-Funktionsbaustein implementiert. Die IEC 61850 Server- und Client-Funktionsbausteine greifen während der Verbindungsherstellung über diese Schnittstelle auf die gespeicherte TLS-Konfiguration und damit auch auf die privaten Schlüssel-, Zertifikatdateien zu. Beim Erfolg wird eine sichere Kommunikationsverbindung hergestellt.

Format der Private-Keys und Zertifikate

Die Zertifikatdateien müssen im PEM-Format (Privacy-Enhanced Mail) vorliegen. Damit der Client oder Server auf die Schlüssel- und Zertifikatdateien problemlos zugreifen kann, wird für die Dateien als Ablageort z.B. folgender Ordner auf dem Windows-Zielsystem empfohlen: `\TwinCAT\3.1\Target\Certificates\IEC61850`.

Bei den privaten Schlüsseldateien und Zertifikatdateien handelt es sich um Textdateien. Eine private Schlüsseldatei beginnt mit der Zeile:

```
„-----BEGIN PRIVATE KEY-----“.
```

Danach folgt der Schlüssel und endet mit der Zeile:

```
„-----END PRIVATE KEY-----“.
```

Beispiel:

```
-----BEGIN PRIVATE KEY-----
MIIEZAIBADANBgkqhkiG9w0BAQEFAASCBCkcgSjAgEAAoIBAQDIx4bVTxEHDhuKWl0XJAKLZfqq
NvMkD26sv/VViprpeMCbU/fSun+2oJO1iczd7Ut66SQBBEPxsFEQhfAh00TaGDDrff9WP7WbkGb7
...
V0um2x+kKJ8hsD9vfORBF7KnwJi0QitwG41PoGDUi6RD8IUybKbLc5a/hx1C5hR0TBnXuxpguzed
pq0NPKFQk7d0ArahQPrYmUySyfagDTALBgNVHQ8xBAMCAJA=
-----END PRIVATE KEY-----
```

Die Zertifikatdateien beginnen mit der Zeile:

```
„-----BEGIN CERTIFICATE-----“.
```

Danach folgen die Zertifikatdaten und enden mit der Zeile:

```
„-----END CERTIFICATE-----“.
```

Beispiel:

```
-----BEGIN CERTIFICATE-----
MIIDyDCCArCgAwIBAgIQRGxL9tvn64FPEmnMEI8FLjANBgkqhkiG9w0BAQsFADB1
MWMwYQYDVQQDDFpSb290IENBLE9VPVR3aW5DQVQsTz1CZWNraG9mZiBBdXRvbWV0
...
6D2L4WEjrzIMR07EWJC4JKvDqxiQMAHsUkpy4vS817ZBsul0/M8EbG9sTdEGxZp
9+GR0Np4ku7TkDJ5
-----END CERTIFICATE-----
```

TLS-Beispielkonfiguration mit dem `FB_IEDTLSecurity`-Funktionsbaustein

Um die TLS-Konfiguration mit Hilfe des Funktionsbausteins `FB_SocketTlsSettingsClass` [► 204] noch weiter vereinfachen zu können, wurde im „ClientWithTLS\Sample30“- und auch im „ServerWithTLS\Sample30“-Projekt, um den `FB_SocketTlsSettingsClass` [► 204]-Funktionsbaustein ein weiterer Hilfs-Funktionsbaustein „`FB_IEDTLSecurity`“ implementiert.

Der „`FB_IEDTLSecurity`“ Funktionsbaustein besitzt nur noch eine Methode: „`Enable`“. Mit Hilfe der „`Enable`“-Methodenparameter kann die TLS-Konfiguration aktiviert/deaktiviert und die Art der TLS-Verschlüsselung in unserem Beispiel festgelegt werden.

```
METHOD FINAL Enable : I_SocketTlsSettingsClass
VAR_INPUT
    bEnable : BOOL;
    bPSK    : BOOL;
END_VAR
```

bEnable: Aktiviert/deaktiviert die TLS-Konfiguration. Bei „`TRUE`“ werden alle notwendigen TLS-Konfigurationseinstellungen in der lokalen Instanz „`fbTls`“ des `FB_SocketTlsSettingsClass` [► 204]-Funktionsbausteins gespeichert. Die TLS-Konfiguration soll in diesem Fall aktiviert werden. Der Wert „`FALSE`“ deaktiviert die TLS-Konfiguration.

bPSK: Konfiguriert die Art der Verschlüsselung. Bei „`TRUE`“ wird eine sichere TLS-Verbindung durch Verwendung eines PSK (Pre-Shared Key) hergestellt. Bei „`FALSE`“ wird eine sichere TLS-Verbindung durch Austausch der Zertifikate hergestellt.

Rückgabeparameter: Wenn ein gültiger Schnittstellenzeiger auf die TLS-Konfigurationseinstellungen zurückgeliefert wird (Rückgabewert <> 0), dann soll eine sichere TLS-Client-Server-Verbindung hergestellt werden. Beim Rückgabewert Null soll eine herkömmliche Verbindung (ohne TLS) hergestellt werden.

Server TLS-Beispielkonfiguration

Die Implementierung der „FB_IEDTLSecurity.Enable“-Methode unterscheidet sich im Client und Server Projekten. Im Serverprojekt werden die Pfade der Zertifikate und der privaten Schlüssel des Servers und im Client-Projekt, die des Clients konfiguriert. Beim Root CA-Zertifikat handelt es sich auf beiden Seiten um das gleiche Zertifikat. Das Root CA-Zertifikat muss auf beiden Systemen vorhanden sein.

```

VAR
  key      : ARRAY[0..14] OF BYTE:=[16#1B,16#D0,16#6F,16#D2,16#56,16#16,16#7D,16#C1,16#E8,16#C7,16#48,16#2A,16#8E,16#F5,16#FF];
  sIdentity : STRING(TCPADS_TLS_PSK_IDENTITY_SIZE) :='MyIdentity';
  sCaPath   : STRING(TCPADS_TLS_CERTIFICATE_PATH_SIZE):='C:\TwinCAT\3.1\Target\Certificates\IEC61850\rootCA.pem';
  sCertPath : STRING(TCPADS_TLS_CERTIFICATE_PATH_SIZE):='C:\TwinCAT\3.1\Target\Certificates\IEC61850\127.0.0.1.pem';
  sKeyPath  : STRING(TCPADS_TLS_CERTIFICATE_PATH_SIZE):='C:\TwinCAT\3.1\Target\Certificates\IEC61850\127.0.0.1.key';
  sKeyPwd   : STRING(TCPADS_TLS_KEY_PASSWORD_SIZE) :='ServerPass';
  sCrlPath  : STRING(TCPADS_TLS_CERTIFICATE_PATH_SIZE):='';
  flags     : ST_TlsListenFlags :=DEFAULT_TLSLISTENFLAGS;
END_VAR

IF bEnable THEN
  Enable:=fbTls.Reset();
  IF bPSK THEN
    fbTls.AddPsk(key:=key, sIdentity:=sIdentity);
  ELSE
    fbTls.AddCa(sCaPath:=sCaPath);
    fbTls.AddCert(sCertPath:=sCertPath, sKeyPath:=sKeyPath, sKeyPwd:=sKeyPwd);
    IF sCrlPath <> '' THEN
      fbTls.AddCrl(sCrlPath:=sCrlPath);
    END_IF
    fbTls.SetListenFlags(flags:=flags);
  END_IF
END_IF

```

Client TLS-Beispielkonfiguration

```

VAR
  key      : ARRAY[0..14] OF BYTE:=[16#1B,16#D0,16#6F,16#D2,16#56,16#16,16#7D,16#C1,16#E8,16#C7,16#48,16#2A,16#8E,16#F5,16#FF];
  sIdentity : STRING(TCPADS_TLS_PSK_IDENTITY_SIZE) :='MyIdentity';
  sCaPath   : STRING(TCPADS_TLS_CERTIFICATE_PATH_SIZE):='C:\TwinCAT\3.1\Target\Certificates\IEC61850\rootCA.pem';
  sCertPath : STRING(TCPADS_TLS_CERTIFICATE_PATH_SIZE):='C:\TwinCAT\3.1\Target\Certificates\IEC61850\client.pem';
  sKeyPath  : STRING(TCPADS_TLS_CERTIFICATE_PATH_SIZE):='C:\TwinCAT\3.1\Target\Certificates\IEC61850\client.key';
  sKeyPwd   : STRING(TCPADS_TLS_KEY_PASSWORD_SIZE) :='ClientPass';
  sCrlPath  : STRING(TCPADS_TLS_CERTIFICATE_PATH_SIZE):='';
  flags     : ST_TlsConnectFlags :=DEFAULT_TLSCONNECTFLAGS;
END_VAR

IF bEnable THEN
  Enable:=fbTls.Reset();
  IF bPSK THEN
    fbTls.AddPsk(key:=key, sIdentity:=sIdentity);
  ELSE
    fbTls.AddCa(sCaPath:=sCaPath);
    fbTls.AddCert(sCertPath:=sCertPath, sKeyPath:=sKeyPath, sKeyPwd:=sKeyPwd);
    IF sCrlPath <> '' THEN
      fbTls.AddCrl(sCrlPath:=sCrlPath);
    END_IF
    fbTls.SetConnectFlags(flags:=flags);
  END_IF
END_IF

```

Initialisierung der TLS-Konfiguration im IEC 61850 Server oder Client

Der „FB_IEDTLSecurity“-Funktionsbaustein wird in der globalen Variablenliste „TcTelecontrol“ instanziiert. Die Übergabe der TLS-Konfigurationseinstellungen an den IEC 61850 Server oder Client erfolgt durch das Setzen der „ipTLS“-Eigenschaft in den Server- oder Client-Protokolleinstellungen. Durch die Zuweisung eines gültigen Schnittstellenzeigers der „ipTLS“-Eigenschaft, werden die TLS-Konfigurationseinstellungen

und damit die TLS-Verschlüsselung aktiviert. Durch die Zuweisung einer Null der „ipTLS“-Eigenschaft, wird die TLS-Konfiguration und die Verschlüsselung deaktiviert. In diesem Beispiel nutzen wir den Rückgabewert der „Enable“-Methode, um die TLS-Verschlüsselung zu aktivieren (Rückgabewert <> Null) oder zu deaktivieren (Rückgabewert = Null). Der Rückgabewert der „Enable“-Methode ist immer Null, wenn der „bEnable“-Methodenparameter „FALSE“ ist.

Setzen der Server TLS-Konfiguration

Der Server in diesem Beispiel besitzt die IP-Adresse: „127.0.0.1“. Diese Adresse muss eventuell angepasst werden.

```
VAR_GLOBAL
...
    fbIEDTLSecurity: FB_IEDTLSecurity;
    fbIEDServer: FB_iec61850ServerClass := (ipIED:=fbIED, settings:=(bEnable:=TRUE, sLocalHost:='127
.0.0.1', ipTLS:=fbIEDTLSecurity.Enable(bEnable:=TRUE, bPSK:=TRUE)));
...
END_VAR
```

Setzen der Client TLS-Konfiguration

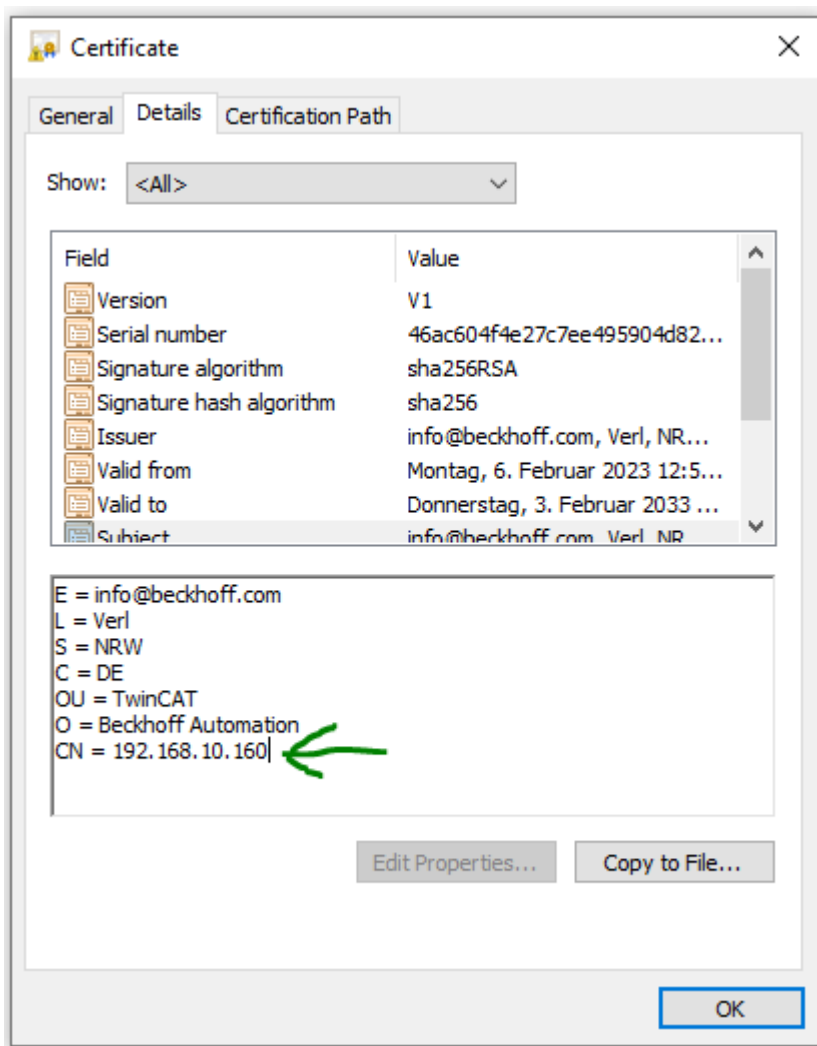
Der Client in diesem Beispiel verbindet sich mit einem Server mit der IP-Adresse: „127.0.0.1“. Auch diese Adresse muss eventuell angepasst werden.

```
VAR_GLOBAL
...
    fbIEDTLSecurity: FB_IEDTLSecurity;
    fbIEDClient: FB_IEDClient := (fbConnection:=(ipIED:=fbIED, settings:=(sRemoteHost:='127.0.0.1',
ipTLS:=fbIEDTLSecurity.Enable(bEnable:=TRUE, bPSK:=TRUE)));
...
END_VAR
```

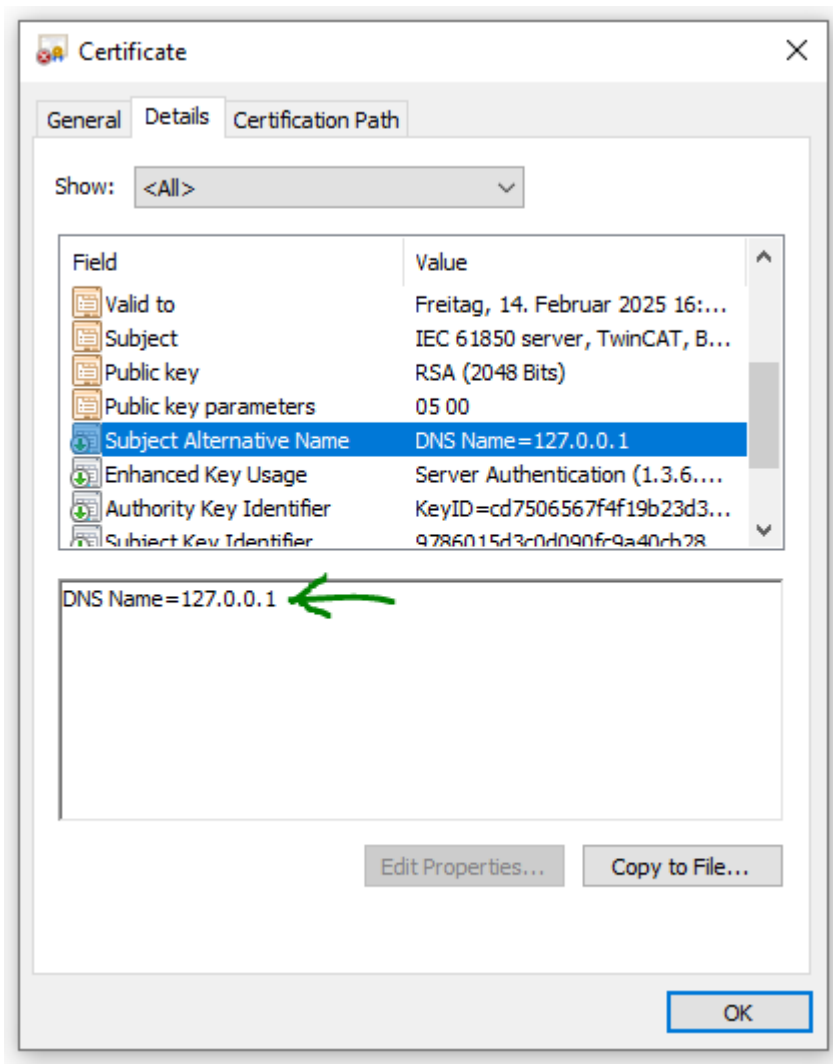
Besonderheiten des Server-Zertifikats

Bei der TLS-Datenverschlüsselung mit Zertifikaten wird der Host-Name oder die IP-Adresse des Servers entweder als „CommonName“- oder als „Subject Alternative Name“-Parameter im Zertifikat konfiguriert. D.h. wenn Sie in diesem Beispiel die IP-Adressen angepasst haben und die TLS-Verschlüsselung mit Zertifikaten testen wollen dann müssen Sie auch den „CommonName“ oder den „Subject Alternative Name“ in Ihrem Zertifikat entsprechend setzen. Zum Testen können selbstsignierte Zertifikate verwendet werden. Solche Zertifikate können z. B. unter Windows mit einem Power Shell-Skript erstellt werden. Unter Windows kann der „CommonName“ oder der „Subject Alternative Name“ relativ einfach überprüft werden.

Beispielzertifikat mit „CommonName“='192.168.10.160':



Beispielzertifikat mit „Subject Alternative Name“ = '127.0.0.1':



Sicherer ISO TP0 Port

Es wird empfohlen, bei aktiven TLS-Kommunikation die von IANA festgelegte (Secure ISO TP0) Portnummer zu verwenden (Standardwert: 3782). Über die Eigenschaft „FB_SocketTlsSettingsClass.nSecurePort“ kann diese Portnummer (für den Server und Client) auf einen anderen Wert umkonfiguriert werden. In unserer Implementierung wird diese Portnummer automatisch verwendet, sobald die TLS-Konfiguration durch das Setzen der „ipTLS“-Eigenschaft im IEC 61850 Client- oder Server-Funktionsbaustein aktiviert wurde. Wenn keine TLS-Konfiguration aktiv ist dann wird beim Server die Portnummer: „fbIEDServer.settings.nLocalPort“ und beim Client die Portnummer: „fbIEDClient.fbConnection.settings.nRemotePort“ verwendet (Standardwert: 102).

Test der sicheren TLS-Verbindung mit PSK

In beiden, Client/Server-Beispielprojekten ist standardmäßig TLS mit PSK (Pre-Shared Key) konfiguriert und aktiviert. Für TLS mit PSK werden keine Zertifikate benötigt. Für einen einfachen Funktionstest müssen nur die IP-Adressen in beiden Projekten angepasst werden.

Test der sicheren TLS-Verbindung mit Zertifikaten

Falls Sie TLS mit Zertifikaten testen möchten, dann kopieren Sie zuerst die Zertifikate und private Schlüssel auf das Server- und Client-Zielsystem.

Server-Zielsystem:

- Root CA-Certificate (z. B.: `C:\TwinCAT\3.1\Target\Certificates\IEC61850\rootCA.pem`)
- Server-Certificate (z. B.: `C:\TwinCAT\3.1\Target\Certificates\IEC61850\server.pem`)

- Privater Schlüssel des Server-Zertifikats (z. B.: C:\TwinCAT\3.1\Target\Certificates\IEC61850\server.key)

Client-Zielsystem:

- Root CA-Certificate (z. B.: C:\TwinCAT\3.1\Target\Certificates\IEC61850\rootCA.pem)
- Client-Certificate (z. B.: C:\TwinCAT\3.1\Target\Certificates\IEC61850\client.pem)
- Privater Schlüssel des Client-Zertifikats (z. B.: C:\TwinCAT\3.1\Target\Certificates\IEC61850\client.key)

Bei den oberen Dateinamen handelt es sich nur um Beispielnamen. Sie können die Pfade und Dateinamen in der „Enable“-Methode des „FB_IEDTLSecurity“-Funktionsbausteins anpassen. Um die TLS-Konfiguration mit Zertifikaten zu aktivieren, muss der „bPSK“-Parameter in der „FB_IEDTLSecurity.Enable“-Methode auf „FALSE“ gesetzt werden.

TLS mit Zertifikat im Server-Projekt aktivieren:

```
VAR_GLOBAL
...

fbIEDServer: FB_iec61850ServerClass := (ipIED:=fbIED, settings:=(bEnable:=TRUE, sLocalHost:='192.168.10.141', ipTLS:=fbIEDTLSecurity.Enable(bEnable:=TRUE, bPSK:=FALSE)));
...
END_VAR
```

TLS mit Zertifikat im Client-Projekt aktivieren:

```
VAR_GLOBAL
...

fbIEDClient: FB_IEDClient := (fbConnection:=(ipIED:=fbIED, settings:=(sRemoteHost:='192.168.10.141', ipTLS:=fbIEDTLSecurity.Enable(bEnable:=TRUE, bPSK:=FALSE)));
...
END_VAR
```

8 Anhang

8.1 Rückgabewerte

8.2 Binary-Time-LWORD-Formatspezifikation

Die TwinCAT Binary-Time Komponenten werden in LWORD auf folgende Weise codiert:

| Bit | Komponente | Typ | Wertebereich | Beschreibung |
|--------|------------|------|----------------|---|
| 0..31 | timeOfDay | TOD | 0..16#05265BFF | Anzahl der vergangenen Millisekunden seit Mitternacht des aktuellen Tages. „16#05265BFF“=23:59:59 Uhr und 999 ms=86399999 ms. |
| 32..47 | day | WORD | 0..16#FFFF | Anzahl der vergangenen Tage seit dem 1. Januar 1984. |
| 48..63 | reserved | WORD | 0 | Diese Bits werden nicht verwendet. Standardwert: 0. |

Beispiele

Die folgende Tabelle enthält einige LWORD-Zahlenwerte und die Werte nach der Konvertierung in den formatierten Binary-Time-String und den strukturierten Typ: `T_BinaryTime` [► 476].

| LWORD | STRING(26) | T_BinaryTime |
|-----------------|------------------------------|---|
| 16#000000000000 | 'BT#1984-01-01-00:00:00' | STRUCT(timeOfDay:=TOD#00:00:00, day:=0) |
| 16#000000000001 | 'BT#1984-01-01-00:00:00.001' | STRUCT(timeOfDay:=TOD#00:00:00.001, day:=0) |
| 16#2F650270E660 | 'BT#2017-03-21-11:22:33.440' | STRUCT(timeOfDay:=TOD#11:22:33.440, day:=16#2F65) |
| 16#30E805265BFF | 'BT#2018-04-12-23:59:59.999' | STRUCT(timeOfDay:=TOD#23:59:59.999, day:=16#30E8) |

8.3 Binary-Time-String-Formatspezifikation

Die Zeichenfolge eines TwinCAT Binary-Time-Strings hat eine max. Länge von 26 Zeichen und folgenden Aufbau: 'BT#[YYYY-MM-DD-]hh:mm:ss[.n[n[n]]]'

Mit der Konvertierungsfunktion: `String_To_BinaryTime6` [► 285] kann die, als String formatierte Binary-Time-Zeitinformation, leicht in die Komponenten des strukturierten Typs: `T_BinaryTime` [► 476] konvertiert werden. Das Datum wird dabei in die „day“-Komponente (2 Byte) und die Uhrzeit in die „timeOfDay“-Komponente (4 Byte) konvertiert.

Das Datum in der Binary-Time-Zeichenfolge ist optional und muss nicht angegeben werden. Wenn das Datum nicht angegeben wurde dann handelt es sich bei der Zeichenfolge um ein 4 Byte Binary-Time-Format (nur die „timeOfDay“-Komponente). In diesem Fall wird bei der Konvertierung der „day“-Komponente implizit der Wert: 0 zugewiesen (dies entspricht dem Datum: 1984-01-01). Bei einer Zeichenfolge mit Datum handelt es sich um ein 6 Byte Binary-Time-Format („timeOfDay“- und „day“-Komponente).

| Variable | Darstellung | Wertebereich | Beschreibung |
|-------------|-------------|----------------|--|
| Prefix | BT# | 'BT#' | Konstanter Präfix. |
| Year | YYYY | '1984'..'2106' | Jahr (4 Dezimalstellen, optional). |
| Month | MM | '01'..'12' | Monat (2 Dezimalstellen, optional). |
| Day | DD | '01'..'31' | Tag (2 Dezimalstellen, optional). |
| Hour | hh | '00'..'23' | Stunde (2 Dezimalstellen). |
| Minute | mm | '00'..'59' | Minute (2 Dezimalstellen). |
| Second | ss | '00'..'59' | Sekunde (2 Dezimalstellen). |
| Millisecond | n[n[n]] | “,0'..'999' | Millisekunde (0 bis 3 Dezimalstellen, optional). |

Beispiele

Die folgende Tabelle enthält einige Zeichenfolgen und die Werte der Komponenten nach einer Konvertierung in den strukturierten Typ: `T_BinaryTime` [▶ 476].

| STRING(26) | T_BinaryTime |
|------------------------------|---|
| 'BT#1984-01-01-00:00:00' | STRUCT(timeOfDay:=TOD#00:00:00, day:=0) |
| 'BT#2018-03-04-11:12:55.6' | STRUCT(timeOfDay:=TOD#11:12:55.600, day:=16#30C1) |
| 'BT#2018-03-04-11:12:55.678' | STRUCT(timeOfDay:=TOD#11:12:55.678, day:=16#30C1) |
| 'BT#01:02:03' | STRUCT(timeOfDay:=TOD#01:02:03, day:=0) |
| 'BT#01:02:03.5' | STRUCT(timeOfDay:=TOD#01:02:03.500, day:=0) |
| 'BT#01:02:03.567' | STRUCT(timeOfDay:=TOD#01:02:03.567, day:=0) |

8.4 UTC-Time-LWORD-Formatspezifikation

Die TwinCAT UTC-Time Komponenten werden in LWORD auf folgende Weise codiert:

| Bit | Komponente | Typ | Wertebereich | Beschreibung |
|--------|------------------------------|------------------|----------------|---|
| 0..31 | secondSinceEpoch | DT | 0..16#FFFFFFFF | Anzahl der vergangenen, ganzen Sekunden seit dem 1. Januar 1970. |
| 32..55 | fractionOfSecond | T_UINT24 [▶ 479] | 0..16#FFFFFF | Sekundenbruchteil codiert als 24 Bit-Zahl. |
| 56 | Quality.LeapSeconds Known | BIT | 0..1 | Qualität der Schaltsekunden (0:=Unbekannt, 1:=Bekannt). |
| 57 | Quality.ClockFailure | BIT | 0..1 | Qualität des Zeitgebers (0:=Gut, 1:=Fehlerhaft). |
| 58 | Quality.ClockNotSynchronized | BIT | 0..1 | Qualität der Uhrzeitsynchronisierung (0:=Synchron, 1:=Nicht synchron). |
| 59 | Quality.Accuracy0 | BIT | 0..1 | Genauigkeit der Zeitinformation (Präzision des Bruchteils der Sekunde). Die Accuracy-Bits bestimmen die Anzahl der signifikanten Bits in der <code>T_UtcTime</code> [▶ 479]. <code>fractionOfSecond</code> -Komponente. |
| 60 | Quality.Accuracy1 | BIT | 0..1 | |
| 61 | Quality.Accuracy2 | BIT | 0..1 | |
| 62 | Quality.Accuracy3 | BIT | 0..1 | |
| 63 | Quality.Accuracy4 | BIT | 0..1 | |

Die UTC-Time.`fractionOfSecond`-Komponente wird in den 24 Bits des Typs: `T_UINT24` auf folgende Weise codiert:

| Bit | Komponente | Wert |
|--------|---------------------------------------|---------------|
| 0 | <code>fractionOfSecond[0].0</code> | 0.5 s |
| 1 | <code>fractionOfSecond[0].1</code> | 0.25 s |
| 2 | <code>fractionOfSecond[0].2</code> | 0.125 s |
| 3 | <code>fractionOfSecond[0].3</code> | 0.0625 s |
| 4 | <code>fractionOfSecond[0].4</code> | 0.03125 s |
| 5 | <code>fractionOfSecond[0].5</code> | 0.015625 s |
| 6 | <code>fractionOfSecond[0].6</code> | 0.0078125 s |
| 7 | <code>fractionOfSecond[0].7</code> | 0.00390625 s |
| 8 | <code>fractionOfSecond[1].0</code> | 0.001953125 s |
| 9..15 | <code>fractionOfSecond[1].1..7</code> | usw. |
| 16..23 | <code>fractionOfSecond[2].0..7</code> | -/- |

Beispiele

Die folgende Tabelle enthält einige Sekundenbruchteilwerte und die Werte der fractionOfSecond-Arrayelemente nach der Konvertierung.

| Sekundenbruchteil | T_UINT24 | Beschreibung |
|-------------------|----------|--|
| 0 s | [0,0,0] | Alle Bits haben den Wert: 0. |
| 0.75 s | [3,0,0] | Bits 0 und 1 sind gesetzt. Wertberechnung: 0.5 s+0.25 s=0.75 s. |
| 0.875 s | [7,0,0] | Bits 0, 1, 2 sind gesetzt. Wertberechnung: 0.5 s+0.25 s+0.125 s=0.875 s. |
| 0.5625 s | [9,0,0] | Bits 0 und 3 sind gesetzt. Wertberechnung: 0.5 s+0.0625 s=0.5625 s. |

Codierung der Präzision in den UTC-Time.Quality.Accuracy0..4-Bits.

| Präzision | Accuracy0 | Accuracy1 | Accuracy2 | Accuracy3 | Accuracy4 | E_UtcTimeAccuracy |
|--------------------|-----------|-----------|-----------|-----------|-----------|--------------------------|
| 0 Bit | 0 | 0 | 0 | 0 | 0 | Null |
| 1 Bit | 0 | 0 | 0 | 0 | 1 | _01 |
| 2 Bit | 0 | 0 | 0 | 1 | 0 | _02 |
| 3 Bit | 0 | 0 | 0 | 1 | 1 | _03 |
| 4 Bit | 0 | 0 | 1 | 0 | 0 | _04 |
| 5 Bit | 0 | 0 | 1 | 0 | 1 | _05 |
| 6 Bit | 0 | 0 | 1 | 1 | 0 | _06 |
| 7 Bit | 0 | 0 | 1 | 1 | 1 | _07 |
| 8 Bit | 0 | 1 | 0 | 0 | 0 | _08 |
| 9 Bit | 0 | 1 | 0 | 0 | 1 | _09 |
| 10 Bit | 0 | 1 | 0 | 1 | 0 | _10 |
| 11 Bit | 0 | 1 | 0 | 1 | 1 | _11 |
| 12 Bit | 0 | 1 | 1 | 0 | 0 | _12 |
| 13 Bit | 0 | 1 | 1 | 0 | 1 | _13 |
| 14 Bit | 0 | 1 | 1 | 1 | 0 | _14 |
| 15 Bit | 0 | 1 | 1 | 1 | 1 | _15 |
| 16 Bit | 1 | 0 | 0 | 0 | 0 | _16 |
| 17 Bit | 1 | 0 | 0 | 0 | 1 | _17 |
| 18 Bit | 1 | 0 | 0 | 1 | 0 | _18 |
| 19 Bit | 1 | 0 | 0 | 1 | 1 | _19 |
| 20 Bit | 1 | 0 | 1 | 0 | 0 | _20 |
| 21 Bit | 1 | 0 | 1 | 0 | 1 | _21 |
| 22 Bit | 1 | 0 | 1 | 1 | 0 | _22 |
| 23 Bit | 1 | 0 | 1 | 1 | 1 | _23 |
| 24 Bit | 1 | 1 | 0 | 0 | 0 | _24 |
| Unzulässig | 1 | 1 | n | n | n | _25_invalid.._30_invalid |
| Nicht spezifiziert | 1 | 1 | 1 | 1 | 1 | Unspecified |

Beispiele

Folgende Tabelle enthält einige LWORD-Werte und die Werte nach der Konvertierung in einen formatierten Binary-Time-String und den strukturierten Typ: `T_UtcTime` [► 479].

| LWO RD | STRING(39) | T_UtcTime | Beschreibung |
|----------------------|--------------------------------------|---|--|
| 16#0000000000000000 | 'UT#1970-01-01-00:00:00.00000000 0' | STRUCT(0) | Alle Komponenten haben den Wert: 0. Datum: 1. Januar 1970. Uhrzeit: 00:00:00 Uhr. Sekundenbruchteil: 0 ms. Präzision: 0 Bit. |
| 16#C4000000386D4380 | 'UT#2000-01-01-00:00:00.00000000 3' | STRUCT(secondSinceEpoch:=DT#2000-01-01-00:00:00, fractionOfSecond:=[0,0,0], quality:=(ClockNotSynchronized:=1, Accuracy3:=1, Accuracy4:=1)) | Datum: 1. Januar 2000. Uhrzeit: 00:00:00 Uhr. Sekundenbruchteil: 0 ms. Zeitgeber ist nicht synchronisiert. Präzision: 3 Bit. |
| 16#A40000001386D4380 | 'UT#2000-01-01-00:00:00.50000000 5' | STRUCT(secondSinceEpoch:=DT#2000-01-01-00:00:00, fractionOfSecond:=[1,0,0], quality:=(ClockNotSynchronized:=1, Accuracy2:=1, Accuracy4:=1)) | Datum: 1. Januar 2000. Uhrzeit: 00:00:00 Uhr. Sekundenbruchteil: 500 ms. Zeitgeber ist nicht synchronisiert. Präzision: 5 Bit. |
| 16#81000000FFF0 | 'UT#2106-02-07-06:28:15.00000000 1' | STRUCT(secondSinceEpoch:=DT#2106-02-07-06:28:15, fractionOfSecond:=[0,0,0], quality:=(LeapSecondsKnown:=1, Accuracy4:=1)) | Datum: 7. Februar 2106. Uhrzeit: 06:28:15 Uhr. Sekundenbruchteil: 0 ms. Schaltsekunden sind unbekannt. Präzision: 1 Bit. |
| 16#C20000007FF0 | 'UT#2106-02-07-06:28:15.87500000 3' | STRUCT(secondSinceEpoch:=DT#2106-02-07-06:28:15, fractionOfSecond:=[7,0,0], quality:=(ClockFailure:=1, Accuracy3:=1, Accuracy4:=1)) | Datum: 7. Februar 2106. Uhrzeit: 06:28:15 Uhr. Sekundenbruchteil: 875 ms. Zeitgeber meldet einen Fehler. Präzision: 3 Bit. |
| 16#1C0000007FF0 | 'UT#2106-02-07-06:28:15.87500000 24' | STRUCT(secondSinceEpoch:=DT#2106-02-07-06:28:15, fractionOfSecond:=[7,0,0], quality:=(ClockNotSynchronized:=1, Accuracy0:=1, Accuracy1:=1)) | Datum: 7. Februar 2106. Uhrzeit: 06:28:15 Uhr. Sekundenbruchteil: 875 ms. Zeitgeber ist nicht synchronisiert. Präzision: 24 Bit. |

8.5 UTC-Time-Quality-BYTE-Formatspezifikation

Die TwinCAT UTC-Time-Quality-Komponenten werden in BYTE auf folgende Weise codiert:

| Bit | Komponente | Typ | Wertebereich | Beschreibung |
|-----|----------------------|-----|--------------|--|
| 0 | LeapSecondsKnown | BIT | 0..1 | Qualität der Schaltsekunden (0:=Unbekannt, 1:=Bekannt). |
| 1 | ClockFailure | BIT | 0..1 | Qualität des Zeitgebers (0:=Gut, 1:=Fehlerhaft). |
| 2 | ClockNotSynchronized | BIT | 0..1 | Qualität der Uhrzeitsynchronisierung (0:=Synchron, 1:=Nicht synchron). |
| 3 | Accuracy0 | BIT | 0..1 (MSB) | Genauigkeit der Zeitinformation (Präzision des Bruchteils der Sekunde). Max. Anzahl der signifikanten Bits in der <u>T_UtcTime</u> [► 479].fractionOfSecond-Komponente. In der Präzision ist Accuracy0 das höchstwertige Bit und Accuracy4 das niedrigstwertige Bit. |
| 4 | Accuracy1 | BIT | 0..1 | |
| 5 | Accuracy2 | BIT | 0..1 | |
| 6 | Accuracy3 | BIT | 0..1 | |
| 7 | Accuracy4 | BIT | 0..1 (LSB) | |

Codierung der Präzision in den UTC-Time.Quality.Accuracy0..4-Bits.

| Präzision | Accuracy0 | Accuracy1 | Accuracy2 | Accuracy3 | Accuracy4 | E_UtcTimeAccuracy |
|--------------------|-----------|-----------|-----------|-----------|-----------|--------------------------|
| 0 Bit | 0 | 0 | 0 | 0 | 0 | Null |
| 1 Bit | 0 | 0 | 0 | 0 | 1 | _01 |
| 2 Bit | 0 | 0 | 0 | 1 | 0 | _02 |
| 3 Bit | 0 | 0 | 0 | 1 | 1 | _03 |
| 4 Bit | 0 | 0 | 1 | 0 | 0 | _04 |
| 5 Bit | 0 | 0 | 1 | 0 | 1 | _05 |
| 6 Bit | 0 | 0 | 1 | 1 | 0 | _06 |
| 7 Bit | 0 | 0 | 1 | 1 | 1 | _07 |
| 8 Bit | 0 | 1 | 0 | 0 | 0 | _08 |
| 9 Bit | 0 | 1 | 0 | 0 | 1 | _09 |
| 10 Bit | 0 | 1 | 0 | 1 | 0 | _10 |
| 11 Bit | 0 | 1 | 0 | 1 | 1 | _11 |
| 12 Bit | 0 | 1 | 1 | 0 | 0 | _12 |
| 13 Bit | 0 | 1 | 1 | 0 | 1 | _13 |
| 14 Bit | 0 | 1 | 1 | 1 | 0 | _14 |
| 15 Bit | 0 | 1 | 1 | 1 | 1 | _15 |
| 16 Bit | 1 | 0 | 0 | 0 | 0 | _16 |
| 17 Bit | 1 | 0 | 0 | 0 | 1 | _17 |
| 18 Bit | 1 | 0 | 0 | 1 | 0 | _18 |
| 19 Bit | 1 | 0 | 0 | 1 | 1 | _19 |
| 20 Bit | 1 | 0 | 1 | 0 | 0 | _20 |
| 21 Bit | 1 | 0 | 1 | 0 | 1 | _21 |
| 22 Bit | 1 | 0 | 1 | 1 | 0 | _22 |
| 23 Bit | 1 | 0 | 1 | 1 | 1 | _23 |
| 24 Bit | 1 | 1 | 0 | 0 | 0 | _24 |
| Unzulässig | 1 | 1 | n | n | n | _25_invalid.._30_invalid |
| Nicht spezifiziert | 1 | 1 | 1 | 1 | 1 | Unspecified |

Beispiele

Die folgende Tabelle enthält einige BYTE-Zahlenwerte und die Werte nach der Konvertierung in den formatierten UTC-Time-Quality-String und in den strukturierten Typ: T_UtcTimeQuality |▶ 480].

| BYTE | STRING(9) | T_UtcTimeQuality | Beschreibung |
|------------|-------------|---|--|
| 2#00000000 | 'UQ#000 0' | STRUCT(0) | Alle Komponenten haben den Wert: 0. Präzision: 0 Bit. |
| 2#11000000 | 'UQ#000 3' | STRUCT(Accuracy3:=1, Accuracy4:=1) | Präzision: 3 Bit. Alle anderen Komponenten haben den Wert 0. |
| 2#10100100 | 'UQ#001 5' | STRUCT(ClockNotSynchronized:=1, Accuracy2:=1, Accuracy4:=1) | Zeitgeber ist nicht synchronisiert. Präzision: 5 Bit. Alle anderen Komponenten haben den Wert 0. |
| 2#11111010 | 'UQ#010 31' | STRUCT(ClockFailure:=1, Accuracy0:=1, Accuracy1:=1, Accuracy2:=1, Accuracy3:=1, Accuracy4:=1) | Zeitgeber meldet einen Fehler. Präzision: Nicht spezifiziert. Alle anderen Komponenten haben den Wert 0. |

8.6 UTC-Time-Quality-String-Formatspezifikation

Die Zeichenfolge eines TwinCAT UTC-Time-Quality-Strings hat folgenden Aufbau: 'UQ#LFC|A[A]'.

| Variable | Darstellung | Wertebereich | Beschreibung |
|----------------------|-------------|--------------|---|
| Prefix | UQ# | 'UQ#' | Konstanter Präfix. |
| LeapSeconds Known | L | '0'..'1' | Qualität der Schaltsekunden ('0':=Unbekannt, '1':=Bekannt). |
| ClockFailure | F | '0'..'1' | Qualität des Zeitgebers ('0':=Gut, '1':=Fehlerhaft). |
| ClockNotSynchronized | C | '0'..'1' | Qualität der Uhrzeitsynchronisierung ('0':=Synchron, '1':=Nicht synchron). |
| Accuracy | A[A] | '0'..'31' | Genauigkeit der Zeitinformation (Präzision des Bruchteils der Sekunde). Max. Anzahl der signifikanten Bits in der <code>T.UtcTime</code> [▶ 479].fractionOfSecond-Komponente (1 oder 2 Dezimalstellen). |

Beispiele

Die folgende Tabelle enthält einige Zeichenfolgen und die Werte der Komponenten nach einer Konvertierung in den strukturierten Typ: `T.UtcTimeQuality` [[▶ 480](#)].

| STRING(9) | T.UtcTimeQuality | Beschreibung |
|-------------|---|---|
| 'UQ#000 0' | STRUCT(0) | Alle Komponenten haben den Wert: 0. |
| 'UQ#001 3' | STRUCT(ClockNotSynchronized:=1, Accuracy3:=1, Accuracy4:=1) | Zeitgeber nicht synchronisiert. Präzision: 3 Bit. Alle anderen Komponenten haben den Wert: 0. |
| 'UQ#000 24' | STRUCT(Accuracy0:=1, Accuracy1:=1) | Präzision: 24 Bit. Alle anderen Komponenten haben den Wert: 0. |

8.7 UTC-Time-String-Formatspezifikation

Die Zeichenfolge eines TwinCAT UTC-Time-Strings hat folgenden Aufbau: 'UT#YYYY-MM-DD-hh:mm:ss.nnnnnnnn|LFC|A[A]'.

| Variable | Darstellung | Wertebereich | Beschreibung |
|----------------------|-------------|-------------------------|---|
| Prefix | UT# | 'UT#' | Konstanter Präfix. |
| Year | YYYY | '1970'..'2106' | Jahr (4 Dezimalstellen). |
| Month | MM | '01'..'12' | Monat (2 Dezimalstellen). |
| Day | DD | '01'..'31' | Tag (2 Dezimalstellen). |
| Hour | hh | '00'..'23' | Stunde (2 Dezimalstellen). |
| Minute | mm | '00'..'59' | Minute (2 Dezimalstellen). |
| Second | ss | '00'..'59' | Sekunde (2 Dezimalstellen). |
| Nanosecond | nnnnnnnn | '000000000'..'99999999' | Nanosekunden: (9 Dezimalstellen). |
| LeapSecondsKnown | L | '0'..'1' | Qualität der Schaltsekunden ('0':=Unbekannt, '1':=Bekannt). |
| ClockFailure | F | '0'..'1' | Qualität des Zeitgebers ('0':=Gut, '1':=Fehlerhaft). |
| ClockNotSynchronized | C | '0'..'1' | Qualität der Uhrzeitsynchronisierung ('0':=Synchron, '1':=Nicht synchron). |
| Accuracy | A[A] | '0'..'31' | Genauigkeit der Zeitinformation (Präzision des Bruchteils der Sekunde). Max. Anzahl der signifikanten Bits in der <code>T.UtcTime</code> [▶ 479].fractionOfSecond-Komponente (1 oder 2 Dezimalstellen). |

Beispiele

Die folgende Tabelle enthält Zeichenfolgen und die Werte der Komponenten nach einer Konvertierung in den strukturierten Typ: `T.UtcTime` [[▶ 479](#)].

| STRING(39) | T_UtcTime | Beschreibung |
|---|---|---|
| 'UT#1970-01-01-00:00:00.000000000 000 0' | STRUCT(0) | Alle Komponenten haben den Wert: 0. Datum: 1. Januar 1970. Uhrzeit: 00:00:00 Uhr. Sekundenbruchteil: 0 ms. Präzision: 0 Bit. |
| 'UT#2018-01-02-03:04:05.125000000 010 3' | STRUCT(secondSinceEpoch:=DT#2018-01-02-03:04:05, fractionOfSecond:=[4,0,0], quality:=(ClockFailure:=1, Accuracy3:=1, Accuracy4:=1)) | Datum: 2. Januar 2018. Uhrzeit: 03:04:05 Uhr. Sekundenbruchteil: 125000000 ns := 125 ms. Zeitgeber meldet einen Fehler. Präzision: 3 Bit. Alle anderen Komponenten haben den Wert: 0. |
| 'UT#2018-01-02-03:04:05.750000000 001 24' | STRUCT(secondSinceEpoch:=DT#2018-01-02-03:04:05, fractionOfSecond:=[3,0,0], quality:=(ClockNotSynchronized:=1, Accuracy0:=1, Accuracy1:=1)) | Datum: 2. Januar 2018. Uhrzeit: 03:04:05 Uhr. Sekundenbruchteil: 750000000 ns := 750 ms. Zeitgeber ist nicht synchronisiert. Präzision: 24 Bit. Alle anderen Komponenten haben den Wert: 0. |

8.8 Quality-String-Formatspezifikation

Die Zeichenfolge eines TwinCAT Quality-Strings hat folgenden Aufbau: 'Q#vv|ORBSFDCA|s|TB'.

| Variable | Darstellung | Wertebereich | Beschreibung |
|--------------------------------|-------------|-----------------|--------------------------------------|
| Prefix | Q# | 'Q#' | Konstanter Präfix. |
| Validity ▶ 687 | vv | '00','01'..'11' | Gültigkeit. |
| Overflow | O | '0'..'1' | Überlauf. |
| OutOfRange | R | '0'..'1' | Nicht im zulässigen Bereich. |
| BadReference | B | '0'..'1' | Falscher Bezug. |
| Oscillatory | S | '0'..'1' | Oszillierend. |
| Failure | F | '0'..'1' | Misserfolg. |
| OldData | D | '0'..'1' | Daten nicht aktuell. |
| Inconsistent | C | '0'..'1' | Inkonsistent. |
| Inaccurate | A | '0'..'1' | Ungenau. |
| Source ▶ 687 | s | '0'..'1' | Quelle ('0':=Prozess, '1':=Ersetzt). |
| Test | T | '0'..'1' | Test. |
| OperatorBlocked | B | '0'..'1' | Bediener blockiert. |

Codierung der Quality.Validity-Komponenten:

| Wert | Validity0 | Validity1 | STRING(2) | Beschreibung |
|--------------|-----------|-----------|-----------|---------------------|
| Good | 0 | 0 | '00' | Gut (Standardwert). |
| Invalid | 0 | 1 | '01' | Ungültig. |
| Reserved | 1 | 0 | '10' | Reserviert. |
| Questionable | 1 | 1 | '11' | Fraglich. |

Codierung der Quality.Source-Komponente:

| Wert | Source | STRING(1) | Beschreibung |
|-------------|--------|-----------|--|
| Process | 0 | '0' | Wert kommt vom Prozess (Standardwert). |
| Substituted | 1 | '1' | Wert wurde ersetzt. |

Beispiele

Die folgende Tabelle enthält Zeichenfolgen und die Werte der Komponenten nach der Konvertierung in den strukturierten Typ: `ST_AcsiQuality` [▶ 463](#).

| STRING(18) | ST_AcsiQuality | Beschreibung |
|----------------------|---|--|
| 'Q#00 00000000 0 00' | STRUCT(0) | Alle Komponenten haben den Wert: 0. |
| 'Q#01 00000000 1 10' | STRUCT(Validity0:=0, Validity1:=1, Source:=1, Test:=1) | Gültigkeit: Ungültig. Quelle: Ersetzt. Test ist gesetzt. Alle anderen Komponenten haben den Wert: 0. |
| 'Q#11 11000001 0 01' | STRUCT(Validity0:=1, Validity1:=1, Overflow:=1, OutOfRange:=1, Inaccurate:=1, OperatorBlocked:=1) | Gültigkeit: Fraglich. Overflow, OutOfRange, Inaccurate und OperatorBlocked sind gesetzt. Alle anderen Komponenten haben den Wert: 0. |

8.9 Quality-WORD-Formatspezifikation

Die TwinCAT Quality-Komponenten werden im WORD auf folgende Weise codiert:

| Bit | Komponente | Typ | Wertebereich | Beschreibung |
|--------|-------------------------------------|-----|--------------|--------------------------------------|
| 0 | Validity0 [▶ 688] | BIT | 0..1 | Gültigkeit MSB. |
| 1 | Validity1 [▶ 688] | BIT | 0..1 | Gültigkeit LSB. |
| 2 | Overflow | BIT | 0..1 | Überlauf. |
| 3 | OutOfRange | BIT | 0..1 | Nicht im zulässigen Bereich. |
| 4 | BadReference | BIT | 0..1 | Falscher Bezug. |
| 5 | Oscillatory | BIT | 0..1 | Oszillierend. |
| 6 | Failure | BIT | 0..1 | Misserfolg. |
| 7 | OldData | BIT | 0..1 | Daten nicht aktuell. |
| 8 | Inconsistent | BIT | 0..1 | Inkonsistent. |
| 9 | Inaccurate | BIT | 0..1 | Ungenau. |
| 10 | Source [▶ 688] | BIT | 0..1 | Quelle ('0':=Prozess, '1':=Ersetzt). |
| 11 | Test | BIT | 0..1 | Test. |
| 12 | OperatorBlocked | BIT | 0..1 | Bediener blockiert. |
| 13..15 | Reserved | BIT | 0 | Werden nicht verwendet. |

Codierung der Quality.Validity-Komponenten:

| Wert | Validity0 | Validity1 | STRING(2) | Beschreibung |
|--------------|-----------|-----------|-----------|---------------------|
| Good | 0 | 0 | '00' | Gut (Standardwert). |
| Invalid | 0 | 1 | '01' | Ungültig. |
| Reserved | 1 | 0 | '10' | Reserviert. |
| Questionable | 1 | 1 | '11' | Fraglich. |

Codierung der Quality.Source-Komponente:

| Wert | Source | STRING(1) | Beschreibung |
|-------------|--------|-----------|--|
| Process | 0 | '0' | Wert kommt vom Prozess (Standardwert). |
| Substituted | 1 | '1' | Wert wurde ersetzt. |

Beispiele

Die folgende Tabelle enthält Zeichenfolgen und die Werte der Komponenten nach der Konvertierung in den strukturierten Typ: [ST_AcsiQuality](#) [[▶ 463](#)].

| WORD | STRING(18) | ST_AcsiQuality | Beschreibung |
|----------------------------|------------------------------|---|---|
| 2#00000
0000000
0000 | 'Q#00
0000000
0 0 00' | STRUCT(0) | Alle Komponenten haben den Wert: 0. |
| 2#00010
0110011
0010 | 'Q#01
0011001
1 0 01' | STRUCT(Validity0:=0, Validity1:=1,
BadReference:=1, Oscillatory:=1,
Inconsistent:=1, Inaccurate:=1, Source:=0,
OperatorBlocked:=1) | Gültigkeit: Ungültig. Quelle: Prozess.
BadReference, Oscillatory, Inconsistent,
Inaccurate, OperatorBlocked sind gesetzt.
Alle anderen Komponenten haben den
Wert: 0. |
| 2#00001
1001100
1101 | 'Q#10
1100110
0 1 10' | STRUCT(Validity0:=1, Validity1:=0,
Overflow:=1, OutOfRange:=1, Failure:=1,
OldData:=1, Source:=1, Test:=1) | Gültigkeit: Reserviert. Quelle: Ersetzt.
Overflow, OutOfRange, Failure, OldData,
Test sind gesetzt. Alle anderen
Komponenten haben den Wert: 0. |

8.10 FAQ - Häufig gestellte Fragen und Antworten

In diesem Bereich werden häufig gestellte Fragen beantwortet, um Ihnen die Arbeit mit TwinCAT IEC 61850 zu erleichtern.

Wenn Sie noch weitere Fragen haben, kontaktieren Sie bitte unseren Support.

Kann mit dem TwinCAT IEC 61850 Client gleichzeitig eine Verbindung zu mehreren Server-Geräten aufgebaut werden? [► 689]

? Kann mit dem TwinCAT IEC 61850 Client gleichzeitig eine Verbindung zu mehreren Server-Geräten aufgebaut werden?

! Ja, das ist möglich. Es muss pro Verbindung eine Instanz des FB_IEC61850ClientClass verfügbar sein.

Mehr Informationen:
www.beckhoff.de/t6510

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

